UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

DETECCIÓN Y ANÁLISIS DE ARTEFACTOS EN LOS PRINCIPALES TIPOS DE CIBERATAQUES

Autor: Sergio García Bordonado
Tutor: Álvaro Manuel Ortigosa Juarez

JUNIO/ 2021

Agradecimientos

En primer lugar quiero agradecerle a mi familia todo el apoyo prestado a lo largo de estos años, sin el cual no habría sido posible afrontar los distintos retos que se me han presentado a lo largo del grado.

A los compañeros de la carrera, los cuales se han convertido en verdaderos amigos y con los cuales he compartido tantos momentos tan difíciles como de alegría durante estos años.

A David Contreras, mi tutor durante mis prácticas curriculares y extracurriculares en One eSecurity gracias al cual me empecé a interesar por el mundo de la ciberseguridad.

Y a los profesores que durante estos años me han transmitido, no solo su conocimiento, sino también su pasión por las distintas áreas del mundo de la Ingeniería Informática.

Resumen (castellano)

Se aborda un trabajo investigativo de tipo descriptivo y documental, mediante una búsqueda sistemática de referentes teóricos que aporten evidencias relacionadas con las amenazas que se ciernen en la actualidad en el entorno informático. Estas amenazas ponen en riesgo, la seguridad en el uso del Internet y hacen vulnerables los datos que los diversos estados y organizaciones almacenan como soporte de sus actividades. Diariamente surgen sofisticadas maneras de afectar la seguridad informática, mediante el uso de troyanos, malware u otros tipos de virus o el uso recurrente del phishing, lo que plantea retos para contrarrestar de manera eficiente esos intentos de apropiarse de datos valiosos. Por ello resulta importante conocer en detalle, la forma mediante la cual operan estos ataques, con la finalidad de proteger adecuadamente la red y hacerla segura para los millones de usuarios. En este contexto, se han desarrollado medios de protección mediante la aplicación de protocolos de seguridad implementados por las ciencias forenses digitales, las cuales estudian las vulnerabilidades de la red y pueden abortar o prevenir ataques a la misma. En este contexto, se describen los principales tipos de ataques, o de ciberamenazas y sus modelados, y la forma de enfrentarlos, así como la caracterización de los mismos y los diversos pasos que conforman el ataque desde el reconocimiento hasta las acciones a llevar a cabo sobre los los equipos infectados. Para ello se realiza un análisis de varios modelos de ataque, que permiten analizar el ciberataque. Además se describen los pasos a seguir para analizar un equipo Windows en caso de que se tenga la sospecha o la certeza de que se ha sufrido un ataque. Esta explicación de los puntos más importantes a analizar en un equipo Windows se realiza junto a un ejemplo de análisis de un servidor web infectado. Para dicho análisis se emplea la máquina virtual SIFT de SANS y algunas herramientas más citadas durante el análisis. Con este análisis no solo se prueba que los distintos tipos de ataque y de malware que se mencionan son una amenaza real, sin que también prueba como los un modelo de ataque puede servir para describir y entender un ataque.

Abstract (English)

An investigative work of a descriptive and documentary type is approached, through a systematic search for theoretical references that provide evidence related to the threats that currently loom in the computing environment. These threats put the security of the use of the Internet at risk and make the data that the various states and organizations store as support for their activities vulnerable. Sophisticated ways to affect computer security emerge every day, through the use of Trojans, malware or other viruses, or the recurrent use of phishing, which pose challenges to efficiently counteract these attempts to steal valuable data. For this reason, it is important to know in detail the way in which these attacks operate in order to adequately protect the network and make it safe for millions of users. In this context, means of protection have been developed through the application of security protocols implemented by digital forensic sciences, which study the vulnerabilities of the network and can abort or prevent attacks on it. In this context, the types of attacks, or cyber-threats and their models, and the way to deal with them, as well as their characterization and the various steps that make up the attack (from recognition of the attack to the infected computer) are broadly described. For this, an analysis of some attack models, which allow us to analyze the cyber-attack and take actions to counter it, must be performed. In addition, the steps in order to analyse a Windows computer, which has been suspected or known to have suffered an attack, is described. Along with this explanation, an example of an infected web server analysis is performed. This analysis utilizes the virtual machine SIFT from SANS and other tools which are cited in the analysis. With this analysis it is not only proved that the different kinds of attacks and malware mentioned in the first part of this thesis are a real threat, but it also proves that using attack models can be useful in order to understand and describe an attack.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

1 Introducción

1.1 Motivación

Internet en la sociedad actual es una parte clave de la vida. Usamos Internet en casa, en la oficina y en dispositivos móviles, donde sea que estemos y donde sea que vayamos. Se ha vuelto importante estar conectado a Internet las 24 horas del día, los 7 días de la semana, para vigilar los negocios, mantenerse en contacto con familiares y amigos, y estar al día en noticias de todo el mundo.

Estar conectado no se trata únicamente sobre los avances en la vida o en los negocios, viene con una serie de peligros potenciales, como: sufrir el robo de datos valiosos, perder objetos personales, robo la privacidad, robo la identidad, que infecten alguno de nuestros dispositivos con malware y mucho más. Todos los días la situación empeora en este aspecto, la seguridad de cualquier red legítima está bajo amenaza de forma constante.

Las ciberamenazas son un tema delicado en el mundo de la seguridad en Internet, gobiernos y organizaciones empresariales en todo el mundo están realizando un enorme esfuerzo para asegurar sus datos. Para ello emplean distintas herramientas y técnicas para mantener sus redes en funcionamiento, mientras los atacantes intentan violar seguridad y enviar software malicioso como botnets, virus, troyanos, etc., todo esto para acceder a datos valiosos.

Hoy en día un gran número de investigadores trabaja en el análisis de amenazas de ciberseguridad generando modelos que predigan las amenazas a las que está expuesta una tecnología. Los mecanismos de defensa principalmente se ocupan de: la comprensión de su propia red, naturaleza del atacante, motivo del atacante, métodos de amenaza y debilidad de seguridad de la red para mitigar futuras amenazas. Para comprender la naturaleza de una ciberamenaza, es importante lograr, en una fase previa a la amenaza, modelar dicha amenaza. Esto se puede hacer de forma personalizada según las necesidades de la organización.

A la hora de prevenir posibles ataques es importante comprender las técnicas de modelado de amenazas, explorarlas y validarlas, al igual que las amenazas de ciberseguridad.

Por tanto, es importante analizar la red para identificar la lista de posibles vulnerabilidades, lo cual nos ayudará a hacernos una idea sobre cómo proteger dicha red. Además, los ataques a nuestro sistema, presenta un riesgo significativo, tanto para la red como para nuestros datos, lo cual requiere tomar acciones de manera urgente y necesaria. La utilización adecuada de las técnicas de modelado de ataques proporciona una planificación avanzada anticipada.

El avance tecnológico en los últimos 20 años ha cambiado la forma en que aprendemos, socializamos y hacemos negocios. Y por supuesto también la forma y el objetivo de ciertos delitos. Es aquí donde entra el análisis forense digital.

Encontrar las pruebas digitales que puedan utilizarse para incriminar o exonerar a un sospechoso, no es tarea fácil pero es imprescindible.

1.2 Objetivos

General

Documentación de los principales tipos de amenazas, detección y análisis de artefactos en los principales tipos de ciberataques.

Específicos

- Explicar los distintos tipos de malware y técnicas de ataque empleadas por los criminales.
- Investigar los 3 modelos más comunes de análisis de ciberamenazas.
- Documentación y ejemplificación de los principales análisis a realizar a un equipo Windows infectado.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo I. Introducción.
- Capítulo II. Ciencia forense digital
- Capítulo III. Modelado de ciberataques.
- Capítulo IV. Análisis y conclusiones sobre los ciberataques

2 Estado del arte

2.1 Planteamiento de la solución

Todo lo expuesto en el apartado anterior hace crucial el lograr modelar un ciberataque, con la finalidad de lograr ahorrar tiempo, dinero y otros recursos para una organización. Se utilizan una serie de técnicas de modelado de ataques, con el fin de analizar los ciberataques, a su vez, permitir tener un entendimiento más avanzado sobre los detalles a considerar relevantes a la hora de enfrentar algún posible ciberataque.

Hoy en día, toda una industria ha evolucionado con el propósito de investigar eventos que ocurren en el ciberespacio para incluir incidentes que involucren espionaje internacional y corporativo, violaciones masivas de datos e incluso el ciberterrorismo. Las oportunidades de empleo en este campo se expanden todos los días. Los profesionales en el campo de la ciberseguridad ahora tienen una gran demanda y se espera que tengan múltiples conjuntos de habilidades en áreas tales como: análisis de malware, computación en la nube, redes sociales, y análisis forense de dispositivos móviles.

La solución a estas nuevas amenazas pasa por que todo profesional dedicado a este área tenga conocimientos básicos sobre las diferentes técnicas, pasos y procesos que se deben de desarrollar como parte del análisis forense digital, pero dado el amplio campo que la investigación forense digital conlleva, se requiere de la creación de un proceso estandarizado que permita cubrir los pasos más esenciales para lograr dicho fin, precisamente esto es lo que se desea realizar en el presente proyecto.

2.2 Ciencia forense digital

El análisis forense digital puede definirse como el proceso de preservación, identificación, extracción y documentación de evidencia informática que puede ser utilizada por un tribunal de justicia. Es una ciencia enfocada a encontrar evidencia de medios digitales como un ordenador , teléfono móvil, servidor o red. Proporciona al equipo forense las mejores técnicas y herramientas para resolver casos complicados relacionados con la tecnología digital. Un experto en análisis forense digital ayuda al equipo forense a analizar, inspeccionar, identificar y preservar la evidencia digital que reside en varios tipos de dispositivos electrónicos (S. Rahavan, 2012)

A continuación, procederemos a enumerar los objetivos más esenciales del área forense digital.

- Recuperar, analizar los materiales relacionados con el ataque.
- Identificar las motivaciones del ataque.
- Identificar al atacante.
- Aplicar los procedimientos adecuados para el manejo de las evidencias encontradas.
- Identificar evidencias.
- Analizar el impacto del ataque.
- Elaborar informes forenses describiendo los procesos de la investigación.
- Aplicar la cadena de custodia de las evidencias.

(Cnsd.gob.pe, 2021)

2.2.1 Fases de la ciencia forense digital.

Los siguientes puntos siguen la estructura de las distintas fases de un proceso de análisis forense digital descritos en Análisis Forense Digital escrito por Miguel López Delgado en 2007.

2.2.1.1 Identificación.

Es el primer paso de un análisis forense. El proceso de identificación incluye principalmente procesos como detectar qué evidencia está presente, dónde se almacena y, por último, cómo se almacena.

2.2.1.2 Preservación.

En esta fase, los datos se aíslan, protegen y conservan. Incluye evitar que las personas usen el dispositivo digital para que la evidencia digital no sea alterada.

2.2.1.3 Análisis.

En este paso, los agentes de investigación reconstruyen fragmentos de datos y extraen conclusiones basadas en la evidencia encontrada. Sin embargo, pueden ser necesarias numerosas iteraciones de examen para respaldar una teoría delictiva específica.

2.2.1.4 Documentación.

En este proceso, se debe crear un registro de todos los datos visibles. Ayuda a recrear el escenario del ataque y a revisarlo. Implica la documentación y descripción adecuada del mismo.

2.2.1.5 Presentación.

En este último paso se realiza el proceso de resumen y explicación de conclusiones.
Sin embargo, debe escribirse con una terminología sencilla que cualquier persona pueda entender.
(K. Ryder, 2011; D. Pinto, 2014)

2.2.2 Tipos de ciencias forenses digitales.

2.2.2.1 Ciencia forense de discos.

Se trata de extraer datos de los medios de almacenamiento mediante la búsqueda de archivos activos, modificados o eliminados.

2.2.2.2 Ciencia forense de redes.

Es una subrama de la ciencia forense digital. Está relacionado con el seguimiento y análisis del tráfico de la red para recopilar información importante y pruebas.

2.2.2.3 Ciencia forense inalámbrica.

Es una división de la ciencia forense de redes. El objetivo principal del análisis forense inalámbrico es ofrecer las herramientas necesarias para recopilar y analizar los datos del tráfico de las redes inalámbricas.

2.2.2.4 Ciencia forense de base de datos.

Es una rama de la ciencia forense digital relacionada con el estudio y examen de bases de datos y sus metadatos relacionados.

2.2.2.5 Ciencia forense de malware.

Esta rama se ocupa de la identificación de códigos maliciosos, para estudiar su carga útil, virus, gusanos, etc.

2.2.2.6 Ciencia forense de email.

Se ocupa de la recuperación y el análisis de correos electrónicos, incluidos los correos electrónicos, calendarios y contactos eliminados.

2.2.2.7 Ciencia forense de memoria.

Se trata de recopilar datos de la memoria del sistema (registros del sistema, caché, RAM) en forma sin procesar y luego tallar los datos del volcado sin procesar.

2.2.2.8 Ciencia forense de teléfonos móviles.

Se ocupa principalmente del examen y análisis de dispositivos móviles. Ayuda a recuperar contactos de teléfono y SIM, registros de llamadas, SMS / MMS entrantes y salientes, audio, videos, etc.

(OpenLearn, 2021)

2.2.3 Ventajas de la ciencia forense.

Aquí están los pros y los beneficios que nos ofrece la ciencia forense digital:

- **I.** Asegurar la integridad del sistema informático.
- **II.** Presentar pruebas en el tribunal, que pueden ayudar a identificar al culpable.
- **III.** Ayuda a las empresas a capturar información importante si sus sistemas informáticos o redes se ven comprometidos.
- **IV.** Rastrea eficazmente a los ciberdelincuentes desde cualquier parte del mundo.
- **V.** Ayuda a proteger los datos, productos digitales tanto de organizaciones como de individuos.
- **VI.** Permite extraer, procesar e interpretar la evidencia fáctica, por lo que acredita la acción del ciberdelincuente en el tribunal.

(K. Ryder, 2021)

2.2.4 Desventajas de la ciencia forense.

A continuación, se muestran los principales inconvenientes de utilizar la ciencia forense:

- **I.** Aunque se detecte una evidencia, para presentarla en un juicio se debe demostrar que no existe manipulación alguna.
- **II.** Producir registros electrónicos y almacenarlos es costoso.
- **III.** Los profesionales del derecho deben tener amplios conocimientos informáticos
- **IV.** Necesidad de producir evidencia auténtica y convincente
- **V.** Si la herramienta utilizada para el análisis forense digital no cumple con los estándares especificados, entonces en el tribunal de justicia, la evidencia puede no ser aprobada.
- **VI.** La falta de conocimiento técnico por parte del investigador podría no ofrecer el resultado deseado.

(K. Ryder, 2021)

2.2.5 Ejemplos en los cuales se puede aplicar la ciencia forense.

En los últimos años, organizaciones comerciales han hecho uso de la ciencia forense en alguno de los siguientes casos:

I. Robo de propiedad intelectual.
II. Espionaje industrial.
III. Disputas laborales.
IV. Investigaciones de fraude.
V. Uso inadecuado de Internet y el correo electrónico en el lugar de trabajo.
VI. Asuntos relacionados con falsificaciones.
VII. Investigaciones de quiebras.
VIII. Problemas relacionados con el cumplimiento normativo

(F.J. Ureña, 2015)

2.3 En qué consiste un ciberataque y qué tipos existen

Un ciberataque es un asalto lanzado por ciberdelincuentes que utilizan uno o más ordenadores contra uno o varios ordenadores o redes. Una ciberamenaza puede deshabilitar ordenadores de forma maliciosa, robar datos o usar un ordenador contaminado como punto de lanzamiento para otras amenazas. Los ciberdelincuentes utilizan una variedad de métodos para lanzar una ciberamenaza, que incluyen malware, phishing, ransomware, denegación de servicio, entre otros métodos.

2.3.1 Tipos de ciberamenazas.

2.3.1.1 DoS y DDoS.

DoS (Denial-of-Service, en español denegación de servicio), DDoS (Distributed Denial-of-Service, en español denegación de servicio distribuidos).Una amenaza DoS se concentra en los recursos de un sistema, con la finalidad que dicho sistema no tenga la opción de retornar un mensaje al requerimiento de un servicio. Un ataque de DDoS también tiene como principal blanco los diferentes elementos principales del sistema, con la salvedad de que se inicia desde un gran número de equipos o dispositivos (host), los cuales se encuentran contaminados por software malicioso (malware), dicho software está bajo el control del atacante.

Al contrario que en amenazas diseñadas para garantizar que el agresor logre hacerse con el control del acceso, el rechazo de servicios no garantiza que el agresor obtenga ganancias de forma directa una vez concretada la amenaza. Muchas veces el objetivo del atacante es simplemente bloquear el servicio en cuestión. Aunque, si el recurso atacado es propiedad de de una empresa u organización competencia del atacante, esto podría ocasionar un beneficio al atacante y por tanto servir de motivación para el ataque. Un propósito extra de un agresor DoS puede ser aislar un sistema para poder lanzar una amenaza totalmente diferente (S.M. Poremba, 2017).

2.3.1.1.1 Tipos de DoS y DDoS.

I. Amenaza de inundación TCP SYN.

En este tipo de amenaza, el agresor aprovecha la utilización del espacio de memoria del búfer mientras se desarrolla una fase de conexión del Protocolo de control de transmisión (TCP, Transmission Control Protocol). El equipo del agresor llena la pequeña cola en

ejecución del sistema objetivo con requerimientos de conexión, pero no ofrece un mensaje en respuesta cuando el sistema objetivo envía un mensaje de respuesta a esos requerimientos solicitados. Lo antes mencionado logra que el grupo de trabajo al cual se llega se desgaste durante la realización de la espera del mensaje de respuesta del equipo del agresor, con lo cual se logra el bloqueo del sistema o que el mismo se quede en un estado de inacción para cuando la cola de enlace se llene.

(W. Eddy, 2007)

II. Amenaza de lágrima.

Esta amenaza ataca el reensamblado en paquetes fragmentados TCP/IP (Internet Protocol), haciendo que se coloquen uno encima del otro; el sistema amenazado intenta reformular los paquetes durante el procedimiento, pero no lo logra. El sistema que sufre el ataque se confunde y procede a bloquearse.

Si no se dispone de parches para poder defenderse contra esta amenaza DoS, se recomienda deshabilitar SMBv2 y obstruir los siguientes puntos de conexión: 445 y 139.

III. Amenaza pitufo.

Este ataque se desarrolla utilizando la suplantación de identidad del protocolo de Internet (Internet Protocol) y el protocolo de mensajes de control de Internet (ICMP, Internet Control Message Protocol), para sobrecargar una red con flujo de datos. Este método de ataque se vale de requerimientos de señal ICMP orientadas a direcciones IP de difusión. Estos requerimientos ICMP tienen como punto de partida una dirección de "víctima" elaborada de forma fraudulenta.

(S. Kumar, 2007)

IV. Amenaza del ping de la muerte.

Utiliza paquetes IP para efectuar la acción de ping a un programa objetivo, que posee un tamaño de IP, el cual sobrepasa el límite superior de 65.535 bytes. Los paquetes de IP de esta dimensión no son autorizados, por tanto, el agresor se ve obligado a desmenuzar el paquete IP. Posterior a que el grupo de trabajo el cual es el destino de la información desmenuzada, logra unir el paquete, puede sufrir sobrecarga de búfer y otros excesos.

Los agresores de ping de la muerte se logran obstruir, haciendo uso de dispositivo de muro de fuego, el cual tendrá como finalidad principal la autenticación de las informaciones IP desmenuzadas para ver el tamaño máximo.

V. Botnets.

Las botnets son los múltiples sistemas contaminados con software malware, todo esto bajo la acción de un hacker para desarrollar amenazas DDoS. Dichos bots o programas zombis tienen como principal meta el atacar los sistemas de destino, por lo general se sobrecarga el ancho de banda del sistema de destino, así como también las funciones de procesamiento. Estas agresiones DDoS son difíciles de identificar y hacerles seguimiento, todo esto debido a las diferentes ubicaciones geográficas en las cuales se encuentran los botnets.

(S. M. Poremba, 2017)

2.3.1.2 Ciberamenazade Man-in-the-Middle (MitM).

Una amenaza MitM (Man in the Middle), sucede al momento en que un hacker logra ingresar entre las informaciones que se envían mutuamente, tanto el cliente como el

servidor. A continuación, mencionaremos algunas de las amenazas más comunes en esta categoría:

2.3.1.2.1 Tipos de ciberamenaza de Man-in-the-Middle (MitM).

I. Secuestro de sesión.

En esta amenaza de tipo MitM, el agresor secuestra la sesión existente entre el cliente y el servidor del grupo de trabajo. El dispositivo agresor reemplaza su dirección IP, mientras el servidor continúa la sesión, con lo cual se le da a entender al servidor que se está comunicando con el cliente, cuando en realidad no es así. Por ejemplo, la amenaza podría desarrollarse así:

- **Paso 01:** Un usuario establece conexión con un servidor
- **Paso 02:** El dispositivo del agresor logra apoderarse del cliente.
- **Paso 03:** El agresor logra desconectar al usuario del servidor.
- **Paso 04:** El dispositivo del agresor logra suplantar la dirección IP del usuario, haciendo uso de su propia dirección IP, a su vez que modifica los números de secuencia del usuario.
- **Paso 05:** El dispositivo del agresor continúa con la sesión entre dicho dispositivo y el servidor, con lo cual este último asume que aún la comunicación es entre él y el dispositivo del usuario.

II. Suplantación de IP.

La suplantación de IP tiene como principal finalidad, el persuadir a un grupo de trabajo de que se está enlazando con un equipo seguro y confiable y proporcionar al atacante entrada al sistema. El agresor envía un paquete con la dirección IP de partida de un dispositivo familiar de confianza, y no así su dirección IP personal, todo esto con un único destino, el cual es un host. El host de destino podría aceptar la información y tomar acciones al respecto.

III. Repetición.

Una amenaza de repetición ocurre cuando un agresor logra capturar y almacenar mensajes viejos y luego intentar retransmitirlos posteriormente, todo esto al momento en que intenta suplantar a alguno de los participantes que intervienen en la comunicación.

Este tipo de amenaza se logra mitigar fácilmente controlando el tiempo de la sesión (un número al azar o una serie que se modifica según el tiempo).

(European Union Agency For CyberSecurity, 2021)

2.3.1.3 Ciberamenaza de phishing y spear phishing.

La amenaza de phishing consiste en la acción de transmitir correos electrónicos que parecen ser originados desde una fuente válida, todo esto con la finalidad de obtener información personal o influir en las personas participantes para que hagan algo. Mezcla ingeniería social y artimañas técnicas. Podría implicar una información que se agregó a un correo electrónico que está contaminado con malware en su dispositivo. También podría ser un link a una web ilegítima que puede engañar al usuario, con el fin de que este sin saberlo acceda a descargar malware o entregue su información personal.

Este es un tipo de amenaza muy dirigida. Los agresores de forma paciente investigan los objetivos y generan información de mucha precisión pública y privada. Gracias a esto, se

hace muy complicado el lograr la plena identificación del spear phishing, e incluso aún más cuesta arriba el lograr contrarrestarlo. Entre las formas más sencillas con las que un cibercriminal logra llevar a cabo una amenaza de spear phishing se encuentra la suplantación de identidad de correo electrónico, que es cuando la información de la sección "De" del correo electrónico se falsifica, lo que hace que se perciba como si proviniera de alguien que conoces, como su gestión o su empresa asociada. Otra técnica que usan los criminales para lograr que sus trampas sean aún más creíbles, es la clonación de una web: copian dicha web legítima para engañarlo y así, lograr que este ingrese su información personal que le permite iniciar la sesión.

(F. Tchakounté, V.S.Nyassi, K.P. Udagepola, 2019)

2.3.1.4 Ciberamenaza de programa malicioso.

El programa malicioso logra describirse como programa no esperado que se coloca en su equipo sin que este tenga conocimiento al respecto. Puede unirse a un código legítimo y esparcirse; logra acechar en programas útiles o transmitirse a través de Internet.

2.3.1.4.1 Tipos de ciberamenaza de programa malicioso.

I. Spyware.

El software espía es un tipo de software que se utiliza para obtener información relacionada con los usuarios, sus equipos o sus comportamientos a la hora de navegar en Internet. Efectúa un rastreo de todas sus actividades sin su aprobación y envía los datos a otro dispositivo ubicado fuera de la red. También puede recibir y ubicar otros paquetes malintencionados de Internet.

II. Adware.

Este es un tipo de programa utilizada por las compañías debido al mercadeo; Las publicidades se despliegan al mismo tiempo en que se desarrolla alguna otra aplicación. El adware se puede descargar de forma programada a su equipo durante el proceso de navegación a través de algún portal web y se puede ver a través de mensajes que aparecen de forma inesperada o mediante una hilera que se visualiza de forma no programada en la pantalla del dispositivo.

III. Ransomware.

Es un tipo de programa malicioso que obstruye el acceso a los datos de la víctima y hace la posible publicación o divulgamiento de los mismos o incluso eliminarlos, en muchas ocasiones este secuestro de la información de la víctima se realiza con la intención de pedir dinero a cambio de devolver el acceso a dicha información. Si bien un programa de ordenador de esta índole puede obstruir el sistema operativo de tal forma que no es difícil de lograr deshacer dichos cambios para una persona con las habilidades necesarias para tal fin, el malware utiliza una técnica denominada extorsión viral, que logra codificar los datos de la víctima de una cierta forma que los hace casi imposibles de recuperar sin la llave de descifrado.

IV. Virus de goteo.

Un virus de goteo es un software que es implementado para ubicar programas maliciosos en los dispositivos. En varias ocasiones, el gotero no está contaminado con programas con fines malvados y, en consecuencia, es posible que el software de detección de virus no lo detecte. Un virus de goteo también logra conectarse a Internet y obtener paquetes nuevos del programa que elimina los virus del equipo, que se aloja en un equipo infectado.

V. Virus de gusano.

Los gusanos se separan de los programas maliciosos y no se quedan atacando un equipo o programa concreto, sino que son aplicaciones autónomas que se esparcen a través de grupos de trabajo y dispositivos. Los virus gusano se esparcen comúnmente a través de paquetes que se enlazan a los emails; el virus gusano se inicializa al momento en que el email es visualizado. Un ataque por gusano típico conlleva que el virus de gusano envíe un duplicado de dicho paquete a cada usuario en el email de un dispositivo portador del virus. Además de efectuar acciones criminales, un gusano que se expande por Internet y excede los dispositivos que almacenan los emails, puede originar amenazas de denegación de utilidad en contra de los nodos del grupo de trabajo, bloqueando de esta manera toda la red de trabajo.

VI. Bombas lógicas.

Es una clasificación de programa malintencionado que se enlaza a una aplicación y se activa por una ocurrencia específica, como una condición lógica o una fecha y hora específicas.

VII. Troyano.

Un troyano es un software que se esconde en una aplicación útil y en la mayoría de los casos tiene intenciones maliciosas. Una característica a resaltar que lo ubica en una categoría diferente entre germen y troyanos es que los troyanos no cuentan con la capacidad de expandirse por sí mismos. Adicionalmente de enviar amenazas a un equipo, un troyano puede configurar una entrada posterior que los atacantes pueden aprovechar. Analicemos el siguiente escenario, un troyano puede configurarse para abrir una entrada para que el cibercriminal pueda darle uso posteriormente y llevar a cabo su a.

VIII. Virus sigiloso.

Los virus sigilosos se esconden en las diferentes operaciones del equipo. Hacen esto comprometiendo el programa de detección de programas maliciosos para que la aplicación informe que un determinado sector que cuenta con la presencia del programa maliciosos, no está en dicha situación. Estos programas maliciosos esconden cualquier alteración sin causar cambios en el tamaño del programa contagiado o variaciones en los datos asociados al calendario, así como la hora de la última modificación del programa.

IX. Virus polimórficos.

Estos programas maliciosos se esconden a través de diferentes procesos de codificación y descodificación. El programa malicioso codificado y una máquina de variación asociada son descifrados en una primera instancia por un software que cumple dicha función. El programa malicioso se dedica a contagiar un sector de código. La máquina de variación luego da origen a una nueva secuencia de ejecución de descodificación y el programa malicioso codifica la máquina de variación y una réplica del programa malicioso con una secuencia de comandos, los cuales se relacionan a la nueva secuencia de comandos de desprogramación. El programa codificado de la máquina de variación y el programa malicioso se enlazan a un nuevo código y la secuencia se lleva a cabo nuevamente. Estos programas maliciosos son difíciles de detectar pero poseen un alto nivel de entropía debido a las múltiples alteraciones de su código principal.

X. Virus de infección del programa o de la lista de arranque.

Un programa malicioso de lista de arranque se enlaza a la lista de arranque principal en los discos duros. Cuando se inicia el equipo, mirará la zona de arranque y activará el programa malicioso en la memoria, donde le es posible la expansión a otros discos y equipos.

XI. Virus de infección de archivos.

Los virus que infectan archivos se caracterizan por adjuntarse al código de ejecución, como programas .exe. El programa malicioso se ubica cuando se ejecuta la secuencia de comandos. Otra versión de un infector de archivos se relaciona con un programa desarrollando un programa malicioso asociado a exactamente un nombre similar, pero con la extensión .exe. En consecuencia, cuando se ejecute el programa, se pondrá en funcionamiento el programa malicioso.

XII. Programa malicioso de enmarcado.

Estos programas maliciosos contagian software de escritorio como editores de texto u hojas de cálculo. Los programas maliciosos de enmarcado se adhieren a la sucesión de inicialización de una aplicación. Al momento en que se habilita la aplicación, el programa pone en funcionamiento los comandos, previo al traspaso del hilo principal de ejecución al programa. El programa malicioso se retransmite y se enlaza a otro código en el equipo.

(Ministerio de asuntos económicos y transformación digital, 2020; N. Baliyan. 2017)

2.3.1.5 Ciberamenaza de cumpleaños.

Los ataques de cumpleaños se realizan sobre comandos de tipo hash, los cuales se implementan para validar la autenticidad de un determinado mensaje o programa. Una información evaluada por una función hash, genera un MD (Message Digest), de dimensión no variable, el cual no depende de la dimensión de la información entrante.

Este MD distingue de una manera única la información. La amenaza de cumpleaños hace referencia a la verosimilitud de lograr obtener dos informaciones al azar que dan origen a 2 MD que son idénticos entre sí, siempre y cuando sean analizados por una secuencia de comandos de tipo hash. Si un cibercriminal procesa un MD, al igual que el mismo usuario, entonces el ciberatacante podría suplantar de una forma segura la información emitida por el usuario en cuestión con el suyo, y el destinatario no notaría que se ha producido una suplantación de información.

(Z. Cao, 2008)

2.3.1.6 Ciberamenaza escucha clandestina.

Las amenazas de escucha clandestina se producen mediante la interceptación del flujo del grupo de trabajo. Al escuchar a escondidas, un agresor puede hacer con las contraseñas, números de tarjetas de crédito y otra información de gran valor que un usuario llegar a enviar a través de la red.

2.3.1.6.1 Tipos de ciberamenaza de escucha clandestina.

I. Ciberamenaza de escucha clandestina activa.

Un atacante captura la información de forma activa haciéndose pasar por un dispositivo conocido que forma parte de la red de la víctima, esto tras enviar solicitudes a los equipos dedicados a la difusión.

II. Ciberamenaza de escucha clandestina pasiva.

Un hacker se percata de la información haciendo de ente receptor de la transmisión del mensaje en el grupo de trabajo.

2.3.1.7 Ciberamenaza de secuencia de comandos entre sitios (XSS).

Utiliza elementos de portales web de terceras personas para poner en funcionamiento, secuencias de comandos en la aplicación que se utiliza para poder hacer búsquedas en Internet, todo esto en el ordenador de la víctima o en la aplicación programable. Específicamente, el cibercriminal introduce un paquete, haciendo uso de JavaScript mal intencionado en el banco de data de un portal.

Para el momento en que el usuario hace el requerimiento de una página de dicho sitio web, el portal emite la respuesta, con el paquete del cibercriminal incluido en el HTML, al programa que está utilizando la víctima para hacer la búsqueda, el cual termina poniendo en estado operativo el paquete enviado por el cibercriminal.

2.3.1.8 Ciberamenaza de inyección SQL.

La inyección de SQL (Structured Query Language), es posible en una situación no deseada pero común en las webs cuya información se maneja en bases de datos. Se lleva a cabo en el momento en que un atacante activa una solicitud SQL a la base de datos, por medio de los datos entrantes a través del portal web y esos datos entrantes son procesados. Un ataque de inyección de SQL tiene la capacidad de visualizar información no autorizada de la base de datos, actualizar información de la base de datos, realizar funciones de administración en la base de datos, restaurar la información presente en un destino determinado y, en ciertas ocasiones, generar órdenes al software. (J. Clarke-Salt, 2012)

2.3.1.9 Ciberamenaza de contraseña.

Dado que las contraseñas son la herramienta más utilizada para autenticar a los usuarios en los sistemas informáticos, la obtención de contraseñas es un enfoque de amenaza común y eficaz. El acceso a la contraseña de una persona se puede obtener revisando la zona de trabajo de la víctima, analizando la conexión a la red para leer las contraseñas no cifradas, mediante el uso de ingeniería social, ganando acceso a una base de datos de contraseñas o adivinando directamente. El último enfoque se puede realizar de forma aleatoria o sistemática:

2.3.1.9.1 Tipos de ciberamenaza de contraseña.

I. Amenaza de diccionario.

Se utiliza un diccionario de contraseñas frecuentes para lograr acceder al ordenador y la red de la víctima. Un método consiste en copiar un archivo cifrado en el cual se encuentran las contraseñas y aplicar el mismo cifrado a un diccionario de contraseñas de uso común para después comparar los resultados.

II. Amenaza de fuerza bruta.

Adivinar contraseñas significa usar un enfoque aleatorio probando diferentes contraseñas y esperando que una funcione. Se puede aplicar cierta lógica probando contraseñas relacionadas información personal de la víctima como el nombre, del puesto de trabajo, aficiones u otros datos personales.

(Ministerio de asuntos económicos y transformación digital, (2020))

3 Modelos de ataque

3.1 Modelado de ciberataques.

Modelar un ciberataque, que aún no ha ocurrido, puede ahorrar tiempo, dinero y otros recursos para una organización. Se utilizan una serie de técnicas de modelado de ataques. Para analizar ciberataques. Esta sección se centra en revisar los tres principales modelos de técnicas de ciberataque llamadas: el modelo de diamante, la cadena de muerte y el gráfico de ataques para el modelado de ataques. El modelo diamante se selecciona debido a la simplicidad del modelo, ya que consta de sólo cuatro componentes principales. La cadena de muerte se utiliza como se ha utilizado durante muchos años en los EE.UU. Departamento de Defensa tanto en ciberdefensa como en la batalla campos.

3.1.1 Modelo diamante.

Es uno de los modelos novedosos para ciberanálisis de intrusión descrito en donde un adversario ataca una víctima dependiendo de dos motivaciones clave en lugar de utilizar una serie de pasos como la cadena de muerte o el gráfico de ataque.
Inicialmente, el adversario comienza sin conocimiento de la capacidad de la víctima. Después de analizar la capacidad de una víctima, el adversario puede encontrar que él / ella tiene más capacidad que la víctima para atacar o no. Este modelo es importante cuando se trata de atacantes más avanzados, como aquellos que ya han ganado cierto control sobre la red. El adversario también analiza la infraestructura de sus habilidades técnicas y lógicas para mandar y controlar cualquiera de las redes de la víctima.
El modelo de diamante también está asociado con algunas metas características como: marca de tiempo, fases, resultado, direcciones, metodología y recursos. En caso de ataque, el modelo diamante identifica fases en una marca de tiempo.

3.1.1.1 Componentes del modelo diamante.

3.1.1.1.1 Adversario.

Es un actor (o conjunto de actores) que ataca a una víctima después de analizar su capacidad contra la víctima.

3.1.1.1.2 Víctima.

Es un actor (o conjunto de actores) que resulta atacado por uno o varios adversarios después de que estos analicen la capacidad de defensa con la que cuenta la víctima.

3.1.1.1.3 Infraestructura.

Se refiere a la forma como están interconectados los diferentes dispositivos de hardware entre sí que conforman la red o redes en las cuales se ubica el o las víctimas, así como los diferentes dispositivos que le permiten al adversario ingresar a la red de la víctima y cometer el ciberataque.

3.1.1.1.4 Capacidad.

Se refiere a las diferentes herramientas, bien sea de software o de hardware, con las que cuenta la potencial víctima para poder defenderse de un posible ciberataque originado por el o los adversarios.
(Sergio Caltagirone, Andrew Pendergast, Christopher Betz, 2013)

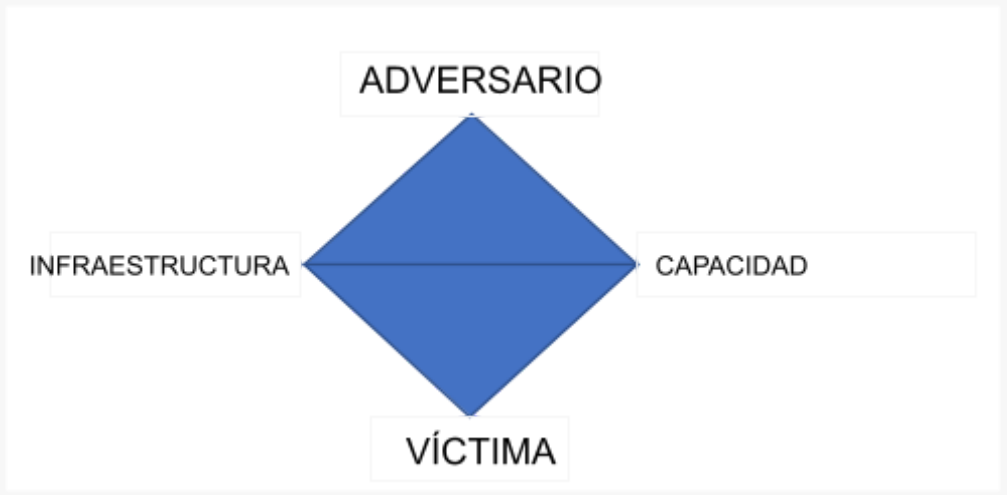

Figura 1. Modelo diamante.

Fuente: Elaboración propia.

3.1.2 Modelo de cadena de muerte.

La cadena de muerte (Kill Chain) para intrusión es una de las técnicas de modelado de ciberataques, que define el ataque como una cadena de acción, es un ataque estructurado, ya que el atacante progresa el ataque en una cadena ordenada según el plan. La cadena de muerte se ha aplicado en otras áreas, incluyendo la ciberseguridad. En ciberseguridad, se utiliza para describir algunos pasos de ataque dentro de un marco de contramedidas.

3.1.2.1 Pasos del modelo de cadena de muerte.

La investigación ha llevado a la cadena de muerte a tener siete pasos de ataque, que se puede describir a continuación:

3.1.2.1.1 Paso 1: Reconocimiento.

El atacante recopila información antes de un ataque. La información se puede recopilar de Internet, que está disponible públicamente.

3.1.2.1.2 Paso 2: Armamento.

El atacante genera una carga útil maliciosa para enviar a la víctima. La carga útil podría ser un virus, un troyano o un archivo ejecutable que pueda realizar alguna acción en la máquina de las víctimas o en la red.

3.1.2.1.3 Paso 3: Entrega.

El atacante envía la carga útil maliciosa a la víctima utilizando algún medio de comunicación. El atacante puede enviar la carga útil por correo electrónico como adjunto o un enlace que descarga la carga útil.

3.1.2.1.4 Paso 4: Explotación.

En esta etapa la explotación real sucede. Si la víctima ha descargado la carga útil en su ordenador comienza la explotación principal. Esta es la etapa en la que el atacante necesita la ayuda de la víctima. Además, esta es una de las fases donde la cadena se puede matar al no descargar la carga útil que envía el adversario.

3.1.2.1.5 Paso 5: Instalación.

Instala malware en los ordenadores infectados u ordenadores de las víctimas. Para infectar el ordenador de la víctima, la carga útil puede necesitar ser ejecutada por la víctima, o se puede ejecutar automáticamente. También es la fase en la que la cadena se puede romper al no ejecutar la carga útil.

3.1.2.1.6 Paso 6: Comando y control.

A través del malware instalado, el atacante crea un comando y canal de control para acceder a los activos internos de la víctima. En esta fase, el atacante ha ganado el control de la máquina de las víctimas.

3.1.2.1.7 Paso 7: Acciones sobre objetivos.

Los atacantes logran su objetivo en el ordenador o red de la víctima que está infectada. Esta podría ser la puerta de entrada del ataque. Los atacantes podrían avanzar hacia datos valiosos de la base de datos a través del servidor web (T. Yadav, 2015).

Figura 2. Modelo de cadena de muerte.

Fuente: Elaboración propia.

3.1.3 Modelo de gráfico de ataque.

Los gráficos de ataque son diagramas conceptuales que se utilizan para analizar cómo un objetivo puede ser atacado. Esto es importante para analizar ciberamenazas en un sistema informático o en una red. Un gráfico de ataque es un gráfico estructurado en árbol, que tiene elementos secundarios de varios niveles con una raíz única. El gráfico o árbol de ataque es una forma tradicional de encontrar la vulnerabilidad, que es introducida por muchas personas incluyendo para desarrollar una herramienta para una defensa eficaz analizando la red.

El gráfico consiste esencialmente en nodos y puede ser de naturaleza compleja cuando se trata de un ataque específico. Puede contener miles de nodos con diferentes caminos. Como nodo central o raíz para este gráfico en forma de árbol se usa el objetivo de dicho

ataque y las diferentes ramas que se desprenden de este están formadas por las distintas maneras que hay de alcanzar dicho objetivo. Estas ramas a su vez también pueden tener subramas al poder considerarse objetivos en sí mismas (Centro Criptológico Nacional, (2012).

4 Análisis y conclusiones sobre los ciberataques

4.1 Ejemplos de análisis de ataque

En este punto se mostrarán tres ejemplos de posibles situaciones en las que un analista de seguridad debe saber qué análisis llevar a cabo para detectar un posible ciberataque y de esta manera ser capáz de, una vez detectado, neutralizar dicho ataque. Dos de estos análisis nos mostrarán pequeños ejemplos, uno de análisis de ficheros y el otro de análisis de tráfico. Por último se incluye un tercer análisis completo que muestra los pasos a seguir durante el análisis forense de un equipo o servidor con sistema operativo windows y prueba la utilidad del uso de modelos para entender cómo se producen los ataques.

4.1.1 Ejemplos de análisis de malware con Yara

A continuación se muestra un ejemplo de detección de un script malicioso, el cual habría sido introducido por el atacante en un equipo poco seguro.
Para la detección de dicho script malicioso vamos a emplear la herramienta de detección de malware YARA.Aunque en este caso hemos elegido emplear esta herramienta hay muchas otras que pueden hacer una función similar.
YARA es es una herramienta de detección de malware la cual utiliza reglas para realizar dicha detección. Estas reglas pueden ser escritas por los investigadores o analistas y pueden estar basadas tanto en patrones de texto plano como de binarios, además de usar reglas lógicas las cuales aplica a los patrones anteriormente mencionados.
A continuación muestro un regla de YARA como ejemplo:

```
 rule ReglaDeEjemplo
 {
meta:
      description = "Esta regla es tan solo un ejemplo"
      author = "Sergio García Bordonado"
strings:
  $a = "Script malicioso"
  $b = { 9B F9 B0 5B C1 83 F0 22 99 6A 4E A9 F7 C0}
condition:
  $a or $b
  }
```

Para esta prueba de análisis de scripts maliciosos se cuenta con un directorio el cual contiene los archivos de una página web.
Para este ejemplo suponemos que el analista está ejecutando YARA con una gran variedad de reglas distintas como parte de un análisis rutinario.
El analista, entre sus distintos ficheros de reglas cuenta con una la cual contiene distintas direcciones URLs las cuales ya han sido utilizadas con anterioridad en otros ataques por algunos cibercriminales para redirigir el tráfico de distintas páginas web a otras controladas por dichos cibercriminales.
A continuación se muestra la estructura del directorio en el que se encuentra la página web:

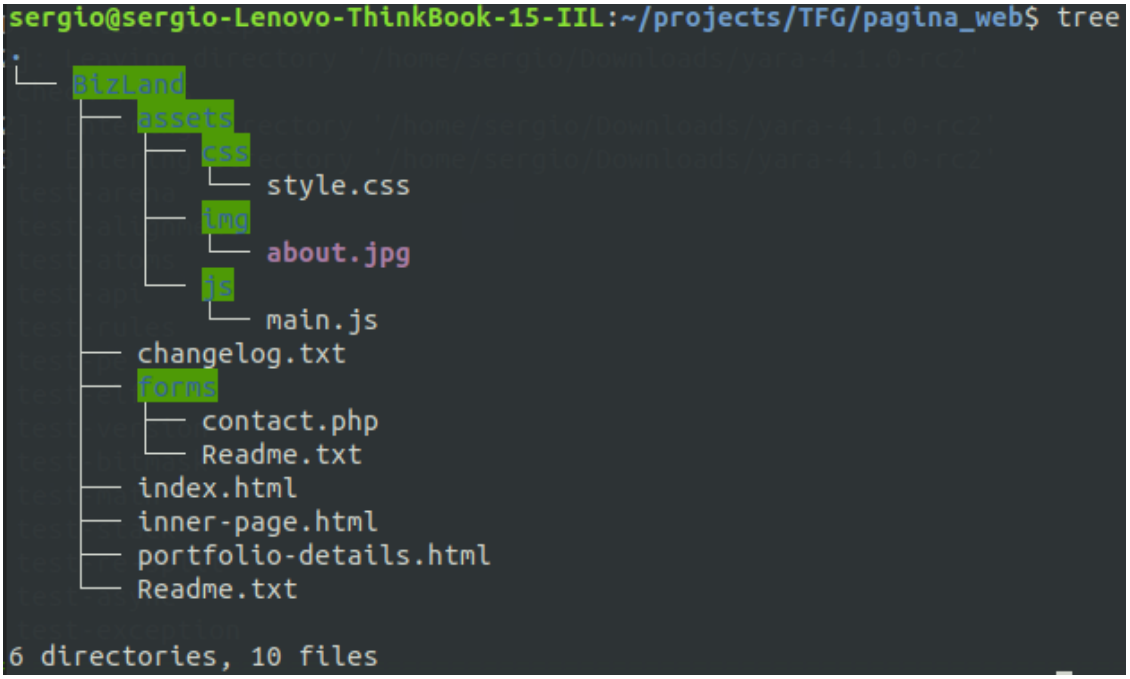

Figura 3. Estructura pagina web

Fuente: Elaboración propia.

Para analizar lso ficheros contenidos dentro del directorio que se muestra en la imagen, el analista utiliza varias reglas de YARA. Entre esta reglas se encuentra una la cual comprobará si se está realizando alguna redirección indevida a alguno de los sitos los cuales se sabe que los hackers suelen utilizar para ectraer información de los usuarios.
La regla en cuestión contiene el siguiente código:

```
rule Regla_URL_Maliciosa
{
        meta:
                description = "Esta regla detecta URLs conocidas por el analista como
maliciosas"
                author = "Sergio García Bordonado"

        strings:
                $a = "sitiomalicioso1"
                $b = "sitiomalicioso2"
                $c = "www.sitiomalicioso3.com"

        condition:
                1 of them
}
```

La regla ha sido creada dentro de el fichero “rule_n1”, el comando para ejecutarla por terminal es el siguiente:

```
yara [OPCIONES] FICHERO_REGLAS FICHERO_A_ANALIZAR
```

En este caso usamos la opción “-s” de YARA para que nos muestre por pantalla que cadena de texto ha hecho saltar la regla:

```
yara -s rule_n1 /home/sergio/projects/TFG/pagina_web/BizLand/assets/js/main.js
```

Como se puede observar en la siguiente imagen, la salida de la execución de la regla YARA nos indica que se ha detectado una de las cadenas maliciosas, en concreto "sitiomalicioso2":

```
sergio@sergio-Lenovo-ThinkBook-15-IIL:~/Downloads/yara-4.1.0-rc2$ yara -s rule_n1
/home/sergio/projects/TFG/pagina_web/BizLand/assets/js/main.js
Regla_URL_Maliciosa /home/sergio/projects/TFG/pagina_web/BizLand/assets/js/main.js
0x1a2f:$b: sitiomalicioso2
```

Figura 4. Ejecución YARA

Fuente: Elaboración propia.

Una vez se ha detectado que se tiene la certeza de que algún atacante ha conseguido modificar el fichero main.js el analista deberá, en primer lugar, preservar de manera correcta la evidencia, en este caso el fichero main.js. Y a continuación revisar el resto de documentos ya que si se ha conseguido detectar en uno podrían haberse añadido más código malicioso en otras partes de la aplicación web.
Después de asegurarse de que todo está correcto y de eliminar la amenaza, se debe centrar en encontrar de qué forma ha logrado acceder al código de la aplicación web el atacante para solucionar dicha brecha de seguridad.
Esta prueba se ha realizado con YARA pero no es ni mucho menos la única que nos permite realizar análisis de malware existen multitud de ellas.

4.1.2 Ejemplos de detección de ataque DOS

Como ya se ha explicado anteriormente, el ataque DOS o ataque de denegación de servicios consiste en impedir el normal funcionamiento del servicio para impedir que sus usuarios accedan al mismo.
Una forma de llevar a cabo este objetivo es, por ejemplo, saturando a la víctima con paquetes UDP. Esta técnica es conocida como UDP flood attack y es la que se va a usar en este ejemplo de detección de ataque DOS.
Para llevar a cabo la detección del ataque DOS es importante monitorear el tráfico que recibe el servidor en el que se aloje el servicio y/o en el que se aloje el cortafuegos si es que lo hay.
En este ejemplo analizamos el tráfico normal que recibe el equipo en el que se encuentra el servidor m. Al analizarlo empezamos a detectar que se está recibiendo lo que parece ser una cantidad inusual de paquetes UDP. La herramienta wireshark nos permite, no solo visualizar el tráfico que recibimos sino también analizarlo, así que la utilizamos para ver de dónde nos están llegando esa cantidad inusual de paquetes.
Para ver de dónde nos llegan esos paquetes primero debemos desplegar las opciones de "statistics" y posteriormente en "Ipv4" seleccionar la opción "Source and Destination address". Al hacerlo se nos abrirá una ventana nueva la cual nos indicará el origen y el destino de los paquetes que llegan a nuestro servidor.

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Source IPv4 Addresses	14054				0,1910	100%	10,0000	10,916
209.126.103.67	10				0,0001	0,07%	0,0400	44,529
192.168.1.152	4				0,0001	0,03%	0,0100	10,217
192.168.1.134	16				0,0002	0,11%	0,0300	43,937
192.168.1.130	14				0,0002	0,10%	0,0100	58,449
127.0.0.53	3				0,0000	0,02%	0,0200	43,946
127.0.0.1	14003				0,1903	99,64%	10,0000	10,916
1.0.0.1	4				0,0001	0,03%	0,0200	5,964
Destination IPv4 Addresses	14054				0,1910	100%	10,0000	10,916
239.255.255.250	9				0,0001	0,06%	0,0100	58,449
224.0.0.251	9				0,0001	0,06%	0,0100	10,217
209.126.103.67	12				0,0002	0,09%	0,0300	44,324
192.168.1.134	14				0,0002	0,10%	0,0400	44,529
127.0.0.53	3				0,0000	0,02%	0,0200	43,937
127.0.0.1	14003				0,1903	99,64%	10,0000	10,916
1.0.0.1	4				0,0001	0,03%	0,0200	5,957

Figura 5. Análisis del origen de los paquetes UDP

Fuente: Elaboración propia.

Como se puede observar en la imagen anterior se está recibiendo una cantidad de tráfico especialmente grande desde una dirección IP en concreto la cual ocupa el 99,64% del tráfico. Esto puede indicar que se está iniciando un ataque de denegación de servicio y habría que proceder a bloquear el tráfico proveniente de dicha dirección IP.

Por lo general la mayoría de los sistemas operativos tienen limitado el número de respuestas ICMP que se generan cuando el paquete UDP llega a un puerto en el que ninguna aplicación está escuchando, el problema de esto es que puede que aquellos paquetes que son legítimos sean ignorados debido a esto.

Aunque pueda parecer que es un ataque que puede mitigarse fácilmente poniendo límites al número de paquetes o mediante el uso de cortafuegos, si el ataque es lo suficientemente grande acabará por saturar los cortafuegos y acabar utilizando el servicio. Además existen múltiples técnicas para burlar estas medidas como puede ser el cambio de las Ips o incluso del MAC desde el cual se envían los paquetes.

En este caso, al tratarse de un ejemplo, el tráfico se recibe desde la IP 127.0.0.1 ya que se ha ejecutado todo desde el mismo equipo. La única diferencia con un caso real sería que el tráfico provendría de un equipo con distinta IP.

4.1.3 Métodos de detección de artefactos en equipos Windows

La mayoría de los ataques citados en el segundo apartado de este trabajo pueden ser detectados si se sabe dónde mirar. Muchos de ellos implican realizar acciones como escalado de privilegios, acceso a otros ficheros, ejecución de malware en el equipo infectado, etc. Todas estas acciones suelen dejar algún tipo de rastro en distintos ficheros del sistema. En este apartado veremos algunos de los análisis que deben hacerse para detectar dichos ataques, en particular en el el caso de los equipos con sistema operativo Windows.

Como ejemplo estos análisis se llevarán a cabo sobre un servidor web con sistema operativo Windows el cual ha sido atacado e infectado. Mediante los ya mencionados análisis trataremos de averiguar, cómo se infectó el equipo y si se ha instalado en él algún tipo de malware, además de todos los rastros y pruebas que podamos recopilar durante el proceso.

El entorno en el que se han realizado todos estos análisis es la máquina virtual SIFT de SANS, una máquina virtual con sistema Ubuntu la cual incluye distintas herramientas para el análisis forense en general, entre ellas todas las que utilizaremos a continuación.

El modelo empleado por el atacante es el de la cadena de muerte. Esto se hará más evidente según vayamos avanzando en el análisis.

4.1.3.1 Análisis de Windows Registry

El registro de Windows consiste en una serie de archivos en los cuales los distintos componentes del sistema y aplicaciones guardan información sobre sus configuraciones y acciones. Además de información sobre las distintas acciones llevadas a cabo por los usuarios.
Es por esto que el registro de Windows es uno de los elementos más importantes a la hora de llevar a cabo el análisis forense de un sistema.
En el siguiente ejemplo de análisis analizaremos algunos de los ficheros más importantes desde el punto de vista del análisis forense presentes en el registro de Windows.
SAM
En este ejemplo de análisis comenzaremos por analizar la información que nos brinda el fichero SAM (Security Account Manager), presente en el directorio "C:\WINDOWS\system32\config\" y que contiene información sobre los distintos usuarios del sistema y sus privilegios.
Para extraer la información del fichero SAM que usaremos para esta prueba usaremos la herramienta RegRipper, y el y el plugin "samparse".
Tras utilizar la herramienta ya mencionada sobre el fichero SAM podemos observar información sobre los distintos usuarios, por ejemplo la información relativa al administrador del equipo. Para ello ejecutamos el siguiente comando desde la dirección en la que tengamos montada la imagen a analizar:

rip.pl -r Windows/System32/config/SAM -p samparse

```
User Information
------------------------
Username        : Administrator [500]
Full Name       :
User Comment    : Built-in account for administering the computer/domain
Account Type    : Default Admin User
Account Created : 2015-08-24 06:54:25Z
Name            :
Last Login Date : 2015-09-12 18:19:18Z
Pwd Reset Date  : 2015-08-24 06:59:37Z
Pwd Fail Date   : 2015-09-02 09:00:39Z
Login Count     : 23
Embedded RID    : 500
  --> Normal user account
```

Figura 6. Análisis fichero SAM, Administrador.

Fuente: Elaboración propia.

Según vamos revisando la información de los distintos usuarios encontramos uno que nos llama la atención, ya que no es un usuario conocido:

```
49 Username        : hacker [1006]
50 Full Name       :
51 User Comment    :
52 Account Type    : Custom Limited Acct
53 Account Created : 2015-09-02 09:05:25Z
54 Name            :
55 Last Login Date : Never
56 Pwd Reset Date  : 2015-09-02 09:05:25Z
57 Pwd Fail Date   : Never
58 Login Count     : 0
59 Embedded RID    : 1006
60   --> Normal user account
```

Figura 7. Análisis fichero SAM, Usuario sospechoso.

Fuente: Elaboración propia.

Además el usuario "hacker" y el usuario "user1" con RIDs 1005 y 1006 están entre los usuarios que pueden conectarse de manera remota, por eso deberíamos tenerlos en cuenta especialmente a la hora de hacer nuestro análisis:

```
91 Group Name    : Remote Desktop Users [2]
92 LastWrite     : 2015-09-02 09:19:24Z
93 Group Comment : Members in this group are granted the right to logon remotely
94 Users :
95   S-1-5-21-3848053756-3249532031-1848221756-1006
96   S-1-5-21-3848053756-3249532031-1848221756-1005
```

Figura 8. Análisis fichero SAM, Grupo de usuarios remotos.

Fuente: Elaboración propia.

Esto aunque podría tratarse de un usuario el cual no conozcamos por algún motivo ya debería llamar nuestra atención y como mínimo deberíamos asegurarnos de que es un usuario válido de nuestro sistema. En caso de que no lo sea deberíamos comprobar rápidamente qué tipo de acciones está llevando a cabo.

SOFTWARE

El fichero SOFTWARE, el cual se encuentra en la dirección "C:\WINDOWS\system32\config\", contiene información sobre la configuración y ajustes de los distintos programas instalados en el equipo y otras del propio sistema Windows. Siempre es interesante analizar este fichero ya que puede darnos pistas sobre qué acciones está llevando a cabo el usuario.

Mediante el uso de la herramienta RegRipper y el con el plugin "apppaths" podemos extraer de este registro los ficheros ".exe" que han sido ejecutados en el equipo. Para esto ejecutamos el siguiente comando desde la dirección en la que hayamos montado la imagen de prueba:

```
rip.pl -r Windows/System32/config/SOFTWARE -p landesk_tln
```

Tras esta ejecución podremos observar lo siguiente:

```
apppaths v.20200511
(NTUSER.DAT,Software) Gets content of App Paths subkeys

2015-08-24 07:49:59Z
  cmmgr32.exe -
  IEXPLORE.EXE - C:\Program Files\Internet Explorer\IEXPLORE.EXE
2008-01-19 11:40:26Z
  install.exe -
  pbrush.exe - %SystemRoot%\System32\mspaint.exe
  setup.exe -
  table30.exe -
  wab.exe - %ProgramFiles%\Windows Mail\wab.exe
  wabmig.exe - %ProgramFiles%\Windows Mail\wabmig.exe
  WORDPAD.EXE - "%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
  WRITE.EXE - "%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
```

Figura 9. Análisis fichero SOFTWARE, Salida apppaths.

Fuente: Elaboración propia.

En la imagen anterior se muestra la salida tras la ejecución del anteriormente citado comando. Como se puede observar se muestra una lista de los ejecutables que se han utilizado. En esta lista en principio no parece haber ningún tipo de comportamiento extraño.

De igual manera que hemos analizado los ejecutables empleados, analizaremos las tareas (Tasks) del sistema. Esto lo haremos porque en ocasiones los atacantes introducen sus propias acciones entre las tareas del sistema con algún fin malicioso. Para ello emplearemos la herramienta RegRipper con el plugin “tasks”. Emplearemos el siguiente comando desde el directorio en el que tengamos montada la imagen:

rip.pl -r Windows/System32/config/SOFTWARE -p tasks > ../../SOFTWARE_tasks.txt

Como hay un gran número de tareas registradas vamos a exportarlas a un fichero txt y después filtrar las que nos interesen. En nuestro caso, como es una imagen de un servidor web que sufrió un ataque en 2015, buscaremos aquellas tareas registradas en 2015. Para ello emplearemos el siguiente comando:

cat SOFTWARE_tasks.txt | grep -B 2 "Reg Time" | grep -B 2 2015

Tras ejecutar el comando podemos observar qué tareas fueron registradas en 2015

```
sansforensics@siftworkstation: ~/Documents/registros
$ cat SOFTWARE_tasks.txt | grep -B 2 "Reg Time" | grep -B 2 2015

Path: \Microsoft\Windows\MUI\LPRemove
Task Reg Time : 2015-08-24 06:54:25Z
```

Figura 10. Análisis fichero SOFTWARE, Salida tasks.

Fuente: Elaboración propia.

Como se puede observar en la ejecución, la única task registrada en 2015 pertenece a "LPRemove", la cual es una de las tasks propias del sistema Windows. Por lo que en principio parece que no se observa ningún comportamiento sospechoso entre las tareas.

NTUSER.DAT

El fichero NTUSER.DAT, el cual se encuentra en el directorio "Windows/System32/config/" es otro de los ficheros más importantes a la hora de analizar los registros de Windows. Esto se debe a que contiene información sobre las configuraciones de los distintos usuarios del sistema

Aunque la imagen con la que estamos trabajando no nos permite acceder a los datos de las apps utilizadas, los documentos abiertos, ni los paths y urls utilizados por los usuarios "user1" y "hacker" ni sus logons ya que no cuenta con carpetas para estos usuarios. Esto se da para este ejemplo en específico, pero en una investigación al uso, siempre que se pueda deberán comprobarse ese tipo de datos mediante el fichero NTUSER.DAT ya que nos pueden dar información clave sobre las acciones que está llevando a cabo el atacante.

4.1.3.2 Logs

Otro tipo de ficheros a tener en cuenta a la hora de analizar cualquier sistema son los logs. Los logs son ficheros en los que quedan registradas las distintas acciones que se van llevando a cabo en el sistema.

Event Logs

Para el caso particular de windows nos centraremos en el análisis de los "Event Logs" o logs de eventos. En estos log es donde quedan registradas todas las acciones llevadas a cabo por el sistema y son cruciales para poder analizar y entender el funcionamiento del mismo. Los "Event Logs" se ubican en la carpeta "Windows/System32/winevt/Logs/". Para esta imagen en particular no hay logs de eventos de windows debido a la propia configuración del sistema. Pero en otros casos en los que sí se disponga de estos logs son una grandísima fuente de información. Los logs de eventos de windows más importantes a la hora de realizar un análisis forense son los siguientes:

- Account Management Events
- Account Logon and Logon Events
- Access to Shared Objects
- Scheduled Tasks Loggins
- Object Access Audit
- Audit Policy Changes
- Audit WIndows Services
- Wireless LAN Auditing
- Process Tracking
- Auditing PowerShell Use

Logs Apache XAMPP

Para este caso en particular al igual que para el de todo aquel equipo que funcione como servidor es muy importante comprobar los los de la propia herramienta que gestiona el servidor. En este caso la herramienta es XAMPP de Apache.

Los logs que estamos buscando se encuentran en el directorio "xampp/apache/logs/". Aquí podremos encontrar los siguientes logs: access.log error.log httpd.pid install.log ssl_request.log.

Como estamos intentando averiguar, entre otras cosas, quién y cómo ha podido acceder al servidor para infectarlo un log que nos puede servir es el "access.log". Este

fichero puede abrirse y leerse sin necesidad de programas específicos, aunque existen programas que facilitan su lectura.
Al abrir el fichero nos encontraremos con mensajes del tipo:

192.168.56.102 - - [23/Aug/2015:15:24:24 -0700] "GET / HTTP/1.1" 302 - "-" "Mozilla/5.0 (X11; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.4.0"

Que indican la IP desde la que llegan las órdenes y la acción realizada. Analizando las distintas acciones registradas en el log es muy fácil darse cuenta de que una gran cantidad de las mismas vienen desde la misma dirección IP, la dirección "192.168.56.102".
Para hacer más sencillo el resto del análisis de este fichero guardaremos todas las acciones llevadas a cabo desde esta IP en un documento a parte mediante el comando:

cat xampp/apache/logs/access.log | grep 192.168.56.102 >> ../../accesslog_192.168.56.102.txt

Ahora que ya tenemos separadas las acciones realizadas desde esta IP sospechosa vamos a analizar sus acciones viendo cuales son las más comunes. Para ello analizaremos el campo "user agent" de los mensajes del log mediante el comando:

cat ../../accesslog_192.168.56.102.txt | awk -F\" '{print $6}' | sort | uniq -c | sort -nr

Este comando nos contará las veces que se repiten lo distintos "user agent" a lo largo del fichero. La salida tras la ejecución del comando es la siguiente:

```
root@siftworkstation:/home/sansforensics/Documents/registros/mounted01/0# cat ../../accesslog_192.16
8.56.102.txt | awk -F\" '{print $6}' | sort | uniq -c | sort -nr
   3621 sqlmap/1.0-dev-nongit-20150902 (http://sqlmap.org)
    510 Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
    215 Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.2.0
     28 Mozilla/5.0 (X11; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.4.0
     22 Mozilla/4.0 (compatible; MSIE 6.0;)
      6 Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.0
```

Figura 11. Análisis Apache Logs, access.log

Fuente: Elaboración propia.

A la vista de los resultados resulta evidente que el atacante ha usado sqlmap para su ataque. Si ha decidido usar esta herramienta lo más probable es que estuviera buscando vulnerabilidades para realizar un ataque de inyección de SQL.
Ahora que ya tenemos una línea más clara desde la que investigar podemos reducir todavía más nuestro área de búsqueda filtrando el fichero que generamos anteriormente y utilizando únicamente aquellas entradas que mencionen "sqlmap".
Aquí una muestra de las acciones llevadas a cabo por el atacante relacionadas con sqlmap.

```
sansforensics@siftworkstation: ~/Documents/registros
$ head accesslog_192.168.56.102_sqlmap.txt | awk -F\" '{print $2}'
GET /dvwa/vulnerabilities/sqli/?id=2&Submit=Submit HTTP/1.1
GET /dvwa/login.php HTTP/1.1
GET /dvwa/vulnerabilities/sqli/?id=2&Submit=Submit&HQMO%3D3319%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%
2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1
GET /dvwa/login.php HTTP/1.1
GET /dvwa/vulnerabilities/sqli/?id=2&Submit=Submit HTTP/1.1
GET /dvwa/login.php HTTP/1.1
GET /dvwa/vulnerabilities/sqli/?id=2%2C%29%28%28%27%28%22%2C%22%27&Submit=Submit HTTP/1.1
GET /dvwa/login.php HTTP/1.1
GET /dvwa/vulnerabilities/sqli/?id=2%27IcSE%3C%27%22%3ExroD&Submit=Submit HTTP/1.1
GET /dvwa/login.php HTTP/1.1
```

Figura 12. Análisis Apache Logs, acciones atacante

Fuente: Elaboración propia.

Como se puede observar están intentando explotar las vulnerabilidades. No solo eso, si seguimos analizando el fichero encontraremos que llegado el momento el atacante empieza a ejecutar comandos utilizando cmd.

```
sansforensics@siftworkstation: ~/Documents/registros
$ cat accesslog_192.168.56.102_sqlmap.txt | awk -F\" '{print $2}' | grep -i cmd
GET /dvwa/vulnerabilities/sqli/?id=2%27%3BSELECT%20BENCHMARK%285000000%2CMD5%280x76446447%29%29%23&Submit=Submit HTTP/1.1
GET /dvwa/vulnerabilities/sqli/?id=2%27%3BSELECT%20BENCHMARK%285000000%2CMD5%280x6a755976%29%29%20AND%20%27GgSr%27%3D%27GgSr&Submit=Submit HTTP/1.1
GET /dvwa/vulnerabilities/sqli/?id=2%27%3BSELECT%20BENCHMARK%285000000%2CMD5%280x627a6b6d%29%29%23&Submit=Submit HTTP/1.1
GET /dvwa/vulnerabilities/sqli/?id=2%27%3BSELECT%20BENCHMARK%285000000%2CMD5%280x54425171%29%29%20AND%20%27DthE%27%3D%27DthE&Submit=Submit HTTP/1.1
GET /tmpbiwuc.php?cmd=echo%20command%20execution%20test HTTP/1.1
GET /tmpbiwuc.php?cmd=dir HTTP/1.1
GET /tmpbiwuc.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpukudk.php HTTP/1.1
GET /tmpbiwuc.php?cmd=del%20%2FF%20%2FQ%20%5Cxampp%5Chtdocs%5Ctmpbiwuc.php HTTP/1.1
GET /tmpbrjvl.php?cmd=echo%20command%20execution%20test HTTP/1.1
GET /tmpbrjvl.php?cmd=del%20%2FF%20%2FQ%20C%3A%5Cxampp%5Chtdocs%5Ctmpudvfh.php HTTP/1.1
GET /tmpbrjvl.php?cmd=del%20%2FF%20%2FQ%20%5Cxampp%5Chtdocs%5Ctmpbrjvl.php HTTP/1.1
```

Figura 13. Análisis Apache Logs, ejecución cmd

Fuente: Elaboración propia.

4.1.3.3 Memoria

A continuación, después de haber descubierto pruebas de los intentos de el ataque vamos a realizar un análisis de la memoria para intentar encontrar más detalles sobre el ataque.
Disponemos de un volcado de la memoria, para analizarlo utilizaremos la herramienta "Volatility".
En primer lugar extraemos la información básica sobre la memoria mediante la ejecución del siguiente comando:

sudo python2 vol.py -f ../Documents/registros/memdump.mem imageinfo

```
sansforensics@siftworkstation: ~/volatility
$ sudo python2 vol.py -f ../Documents/registros/memdump.mem imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : VistaSP1x86, Win2008SP1x86, Win2008SP2x86, VistaSP2x86
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/sansforensics/Documents/registros/memdump.m
em)
                      PAE type : PAE
                           DTB : 0x122000L
                          KDBG : 0x81716c90L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x81717800L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2015-09-03 10:04:05 UTC+0000
     Image local date and time : 2015-09-03 03:04:05 -0700
```

Figura 14. Análisis Memoria, información memoria

Fuente: Elaboración propia.

El análisis de la memoria lo dividiremos en distintos subanálisis:

- Conexiones de Red
- Servicios en ejecución
- Artefactos del registro de Windows en memoria
- Procesos

Conexiones de Red

Es importante analizar las conexiones que nos muestra el volcado de memoria, aunque los datos que aquí podremos encontrar pueden no estar completos debido a que la información de alguna de las conexiones no se encuentre en memoria por motivos de paginado. O incluso que exista algún malware en el momento en el que se realiza el volcado de memoria que esté ocultando algunas de las conexiones.

Como estamos utilizando la herramienta volatility esta vez la ejecutaremos con el plugin de "netscan", el cual extrae información sobre las distintas conexiones establecidas. Ejecutaremos el comando:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 netscan

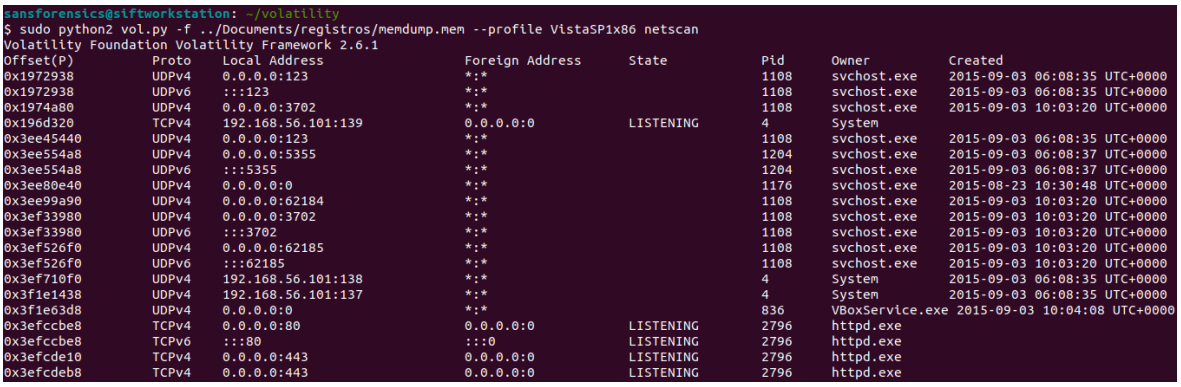

Figura 15. Análisis Memoria, conexiones de red

Fuente: Elaboración propia.

Utilizando este plugin, podemos ver los protocolos, las direcciones y puertos de origen y destino, el estado en que se encuentran, el identificador del proceso que la está ejecutando, el propietario y la fecha de creación de las distintas conexiones. En este caso en particular, o bien porque el atacante no estaba conectado en el momento en el que se hizo el volcado de memoria o bien porque su conexión esté de alguna manera oculta, a excepción de las conexiones con PID 2796 y 2880, las cuales son ejecutadas por "httpd.exe" no parece haber nada fuera de lugar entre las conexiones. Aun así seguiremos analizando el resto de la información que nos da el volcado de la memoria antes de analizar dichos procesos.

Servicios en ejecución

Los servicios son procesos que se ejecutan de forma automática por el sistema sin necesidad de intervención del usuario. En ocasiones los atacantes pueden incluir malware entre estos servicios para hacerlo pasar desapercibido. Habitualmente la forma más sencilla de analizar los servicios en ejecución detectados en el volcado de memoria es compararla con una lista de servicios que de forma habitual si deberían estar en ejecución para el

funcionamiento normal del equipo. Como en este caso en concreto no contamos con una lista de los servicios que suelen estar en ejecución durante el funcionamiento normal del servidor simplemente nos limitaremos a extraer del volcado de memoria los servicios que hayan sido detectados por la herramienta Volatility y analizarlos nosotros mismos por si vemos algo fuera de lugar que llame nuestra atención. Par poder ver los servicios detectados ejecutamos el siguiente comando:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 svcscan

La información sobre cada uno de los servicios se mostrará de la siguiente manera:

```
sansforensics@siftworkstation: ~/volatility
$ sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 svcscan
Volatility Foundation Volatility Framework 2.6.1
Offset: 0xe41888
Order: 28
Start: SERVICE_AUTO_START
Process ID: 1024
Service Name: BITS
Display Name: Background Intelligent Transfer Service
Service Type: SERVICE_WIN32_SHARE_PROCESS
Service State: SERVICE_RUNNING
Binary Path: C:\Windows\system32\svchost.exe -k netsvcs

Offset: 0xe417f0
Order: 27
Start: SERVICE_AUTO_START
Process ID: 1352
Service Name: BFE
Display Name: Base Filtering Engine
Service Type: SERVICE_WIN32_SHARE_PROCESS
Service State: SERVICE_RUNNING
Binary Path: C:\Windows\system32\svchost.exe -k LocalServiceNoNetwork
```

Figura 16. Análisis Memoria, servicios en ejecución

Fuente: Elaboración propia.

En este caso al no tener una lista de servicios habituales no podemos hacer un análisis muy exhaustivo que nos resalte las anomalías con respecto a su funcionamiento normal. Aun así después de analizarlo de forma visual no se ha encontrado ningún servicio que resulte sospechoso.

Artefactos del registro de Windows en memoria

Los distintos archivos de los registros de Windows, de los cuales ya hemos hablado en puntos anteriores de esta explicación y ejemplo sobre los distintos análisis a realizar en sistemas Windows, suelen utilizarse y actualizarse de manera frecuente por el sistema operativo. Es por eso que en ocasiones pueden encontrarse partes de los mismos al hacer el volcado de memoria.

A continuación analizaremos algunos ejemplos de estos ficheros para ver si pueden mostrarnos alguna información sobre el ataque.

Shellbags:

Los “Shellbags” son unos elementos presentes en el el registro de Windows los cuales cotienen información acerca de qué ficheros han sido accedidos o modificados. Es posible acceder a esta información si se encuentra en la memoria en el momento en que se realiza

el volcado de la misma. Para ello, utilizaremos el programa “Volatility” y el plugin “shellbags” que nos permite viasualizar la información contenida en los “shellbags”:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 shellbags

Tras la ejecución, en caso de encontrarse los ficheros cargados en memoria, obtendremos la información de las shellbags como podemos observar en la siguiente imagen:

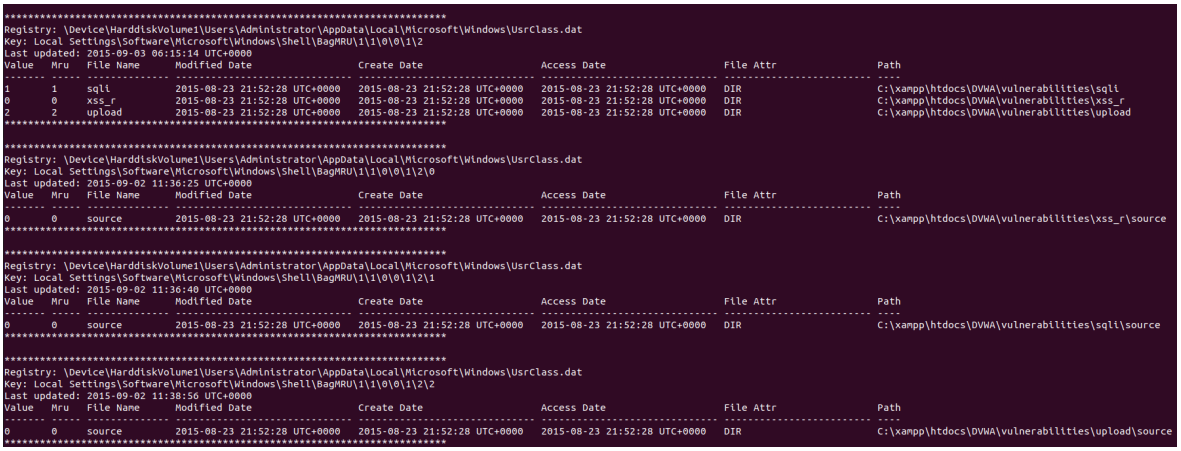

Figura 17. Análisis Memoria, shellbags

Fuente: Elaboración propia.

Como se puede observar en la Figura 17, alguien parece estar explorando las vulnerabilidades de Xampp. En concreto parece que el atacante planea un ataque de inyección sql, de XSS o de subida de archivos. Lo cual coincide con lo visto durante el análisis de los logs de Xampp.

Windows Application Compatibility Database:
El análisis de la memoria, al igual que otros análisis ya realizados, nos permite acceder a la Windows Application Compatibility Database mostrando también en qué momento fueron modificados por última vez
Para poder realizar este análisis utilizaremos la herramienta “Volatility” con el plugin “shimcache”. Para acceder esta información y poder analizarla ejecutamos el siguiente comando:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 shimcache | sort | grep 2015

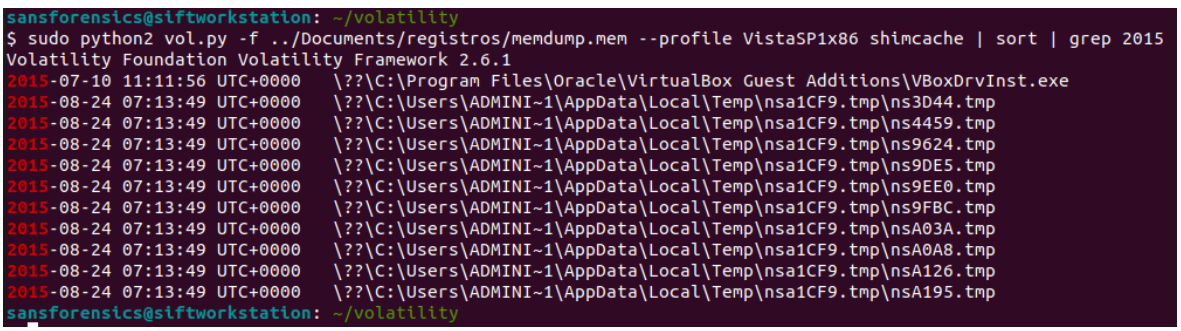

Figura 18. Análisis Memoria, shimcache

Fuente: Elaboración propia.

Claves Registry:
Por último nos permite ir analizando clave a clave la información que encuentre para cada una de ellas. Esto lo haremos mediante la utilización de la herramienta "Volatility" y el plugin "printkey". El comando que debemos ejecutar es el siguiente:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 printkey -K <CLAVE>

```
sansforensics@siftworkstation: ~/volatility
$ sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 printkey -K SOFTWARE
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \Device\HarddiskVolume1\Windows\System32\config\DEFAULT
Key name: Software (S)
Last updated: 2008-01-19 11:36:57 UTC+0000

Subkeys:
  (S) Microsoft
  (S) Policies

Values:
```

Figura 19. Análisis Memoria, claves

Fuente: Elaboración propia.

Este mismo comando se puede emplear para cualquiera de las claves descubiertas. Para obtener la lista de las claves detectadas usaremos la herramienta "Volatility" y el plugin "hivelist". Para obtener dicha lista ejecutaremos el siguiente comando:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 hivelist

```
sansforensics@siftworkstation: ~/volatility
$ sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 hivelist
Volatility Foundation Volatility Framework 2.6.1
Virtual    Physical   Name
---------- ---------- ----
0x87b4ba20 0x3c0c0a20 \Device\HarddiskVolume1\Windows\System32\config\COMPONENTS
0x87b55a20 0x3c192a20 \Device\HarddiskVolume1\Windows\System32\config\SOFTWARE
0x87b7d008 0x3a6a2008 \Device\HarddiskVolume1\Windows\System32\config\SAM
0x87b7d6a8 0x3a6a26a8 \Device\HarddiskVolume1\Windows\System32\config\DEFAULT
0x8ab1aa20 0x3c285a20 \Device\HarddiskVolume1\Boot\BCD
0x8f4dba20 0x25828a20 \Device\HarddiskVolume1\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0x8f565a20 0x251eba20 \Device\HarddiskVolume1\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0x90edca20 0x1c1d5a20 \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\UsrClass.dat
0x90f09a20 0x1ab8ea20 \Device\HarddiskVolume1\Users\Administrator\NTUSER.DAT
0x86210008 0x00ac8008 [no name]
0x86226008 0x00a94008 \REGISTRY\MACHINE\SYSTEM
0x86246008 0x00a76008 \REGISTRY\MACHINE\HARDWARE
0x87b17a20 0x3c1f5a20 \Device\HarddiskVolume1\Windows\System32\config\SECURITY
```

Figura 20. Análisis Memoria, hivelist

Fuente: Elaboración propia.

Procesos
Otra forma de analizar qué está sucediendo en un equipo es analizar sus procesos en ejecución. Los atacantes pueden ejecutar malware con sus propios procesos o inyectando

código en el contexto de otros procesos. Por eso el análisis de procesos es una parte tan importante en una investigación forense.
Para realizar el análisis de los procesos que se hayan detectado dentro del volcado de la memoria utilizaremos la herramienta "Volatility" y el plugin "pslist" el cual nos devolverá una lista con los procesos detectados. Para nuestro caso particular empleamos el siguiente comando:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 pslist

```
sansforensics@siftworkstation: ~/volatility
$ sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------
0x82f57910 System                    4      0    105      504 ------      0 2015-08-23 20:27:20 UTC+0000
0x838382d0 smss.exe                420      4      4       28 ------      0 2015-08-23 20:27:20 UTC+0000
0x83912208 csrss.exe               484    472     11      400      0      0 2015-08-23 20:27:22 UTC+0000
0x8392d530 csrss.exe               524    516      9      536      1      0 2015-08-23 20:27:28 UTC+0000
0x8392c9f8 wininit.exe             532    472      3      102      0      0 2015-08-23 20:27:28 UTC+0000
0x8387ed90 winlogon.exe            560    516      4      125      1      0 2015-08-23 20:27:28 UTC+0000
0x8393bd90 services.exe            608    532      7      238      0      0 2015-08-23 20:29:06 UTC+0000
0x83942020 lsass.exe               620    532     19      628      0      0 2015-08-23 20:29:18 UTC+0000
0x83945d90 lsm.exe                 628    532     10      166      0      0 2015-08-23 20:29:19 UTC+0000
0x839d4020 svchost.exe             792    608      8      305      0      0 2015-08-23 20:29:45 UTC+0000
0x839ded90 VBoxService.exe         836    608      8      115      0      0 2015-08-23 20:29:46 UTC+0000
0x839f0020 svchost.exe             892    608      7      262      0      0 2015-08-23 10:29:52 UTC+0000
0x83a06020 svchost.exe             984    608     15      306      0      0 2015-08-23 10:29:52 UTC+0000
0x83a18020 svchost.exe            1012    608      6      147      0      0 2015-08-23 10:29:53 UTC+0000
0x83a0eb88 svchost.exe            1024    608     37      913      0      0 2015-08-23 10:29:53 UTC+0000
```

Figura 21. Análisis Memoria, procesos

Fuente: Elaboración propia.

Tras ejecutar el comando y observar la lista de procesos nos damos cuenta de que hay dos ejecuciones de la terminal cmd de Windows.

```
sansforensics@siftworkstation: ~/volatility
$ sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 pslist | grep cmd
Volatility Foundation Volatility Framework 2.6.1
0x83e7b7f8 cmd.exe                612    816      1       72      1      0 2015-08-23 10:30:44 UTC+0000
0x84259100 cmd.exe               1972    816      1       19      1      0 2015-09-02 09:28:30 UTC+0000
```

Figura 22. Análisis Memoria, procesos cmd

Fuente: Elaboración propia.
Vamos a investigar qué clase de comandos se estaban ejecutando para ver si están relacionados con el ataque. Para ello utilizaremos la herramienta "Volatility" y el plugin "cmdscan". Para ello ejecutaremos el siguiente comando:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 cmdscan

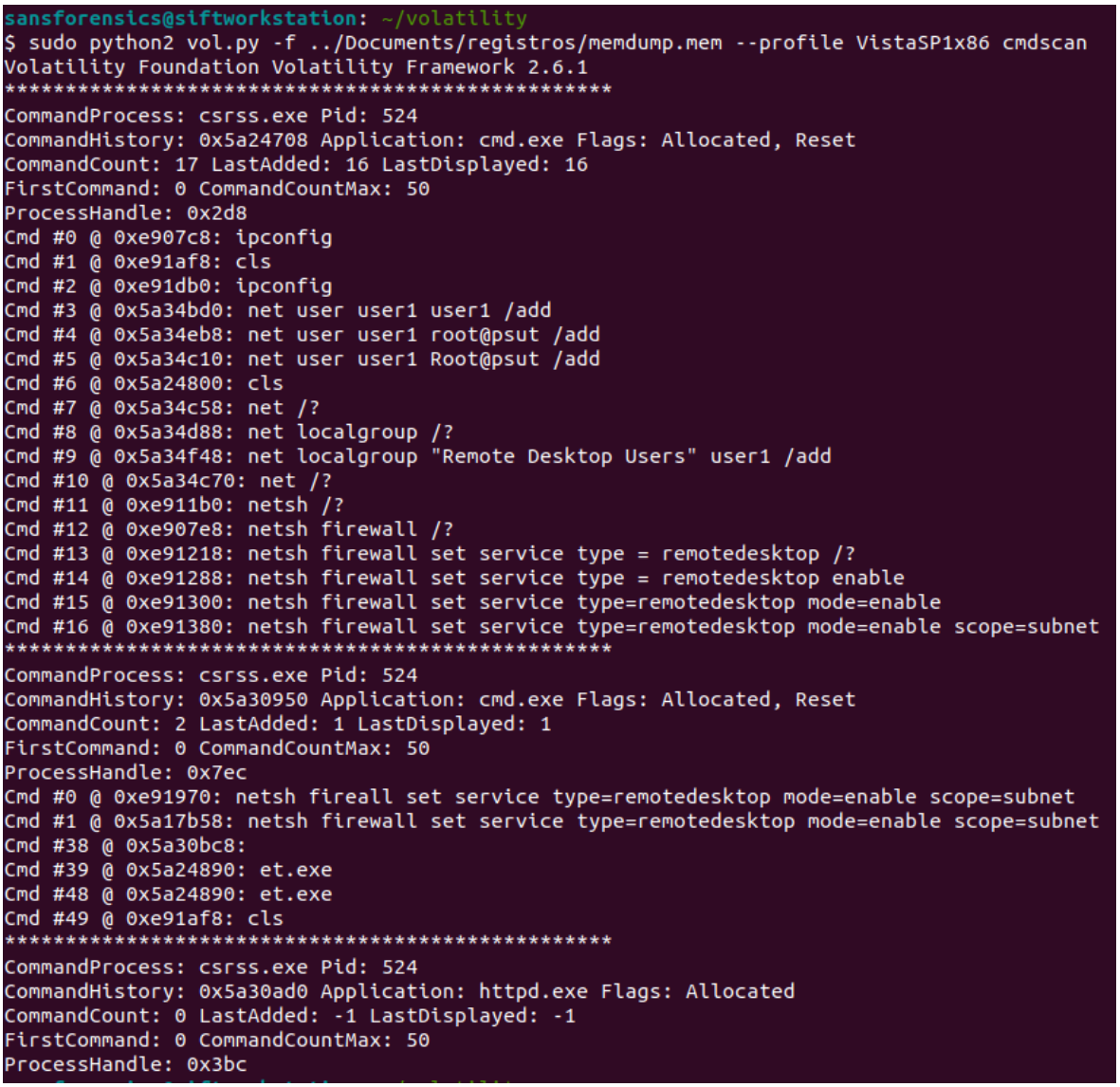

Figura 23. Análisis Memoria, comandos cmd

Fuente: Elaboración propia.

Como se puede observar en la imagen anterior. Se ve que el atacante ha ejecutado "ipconfig" probablemente para conocer la IP. Posteriormente, el atacante ha añadido al usuario "user1" al grupo "Remote Desktop Users" y por último ha cambiado la configuración del firewall para que se permitan las conexiones remotas.

Ahora que ya hemos comprobado cómo se ha añadido al usuario "user1" a la lista de usuarios remotos debemos averiguar cómo se ha añadido al usuario "hacker", el cual según lo que pudimos observar en el análisis de registros de Windows también estaba en dicho grupo.

Para ello analizaremos los procesos sospechosos que detectamos al analizar las conexiones, que ejecutaban httpd.exe. Estos procesos tenían los PID 2880 y 2796 como ya pudimos observar.

Para volcar la información de los procesos en un fichero y poder analizarla utilizaremos los siguientes comandos:

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 memdump -p 2796 --dump-dir ../Documents/registros/dump2796/

sudo python2 vol.py -f ../Documents/registros/memdump.mem --profile VistaSP1x86 memdump -p 2880 --dump-dir ../Documents/registros/dump2880/

A continuación buscamos información sobre el usuario "hacker" en el volcado de los procesos.

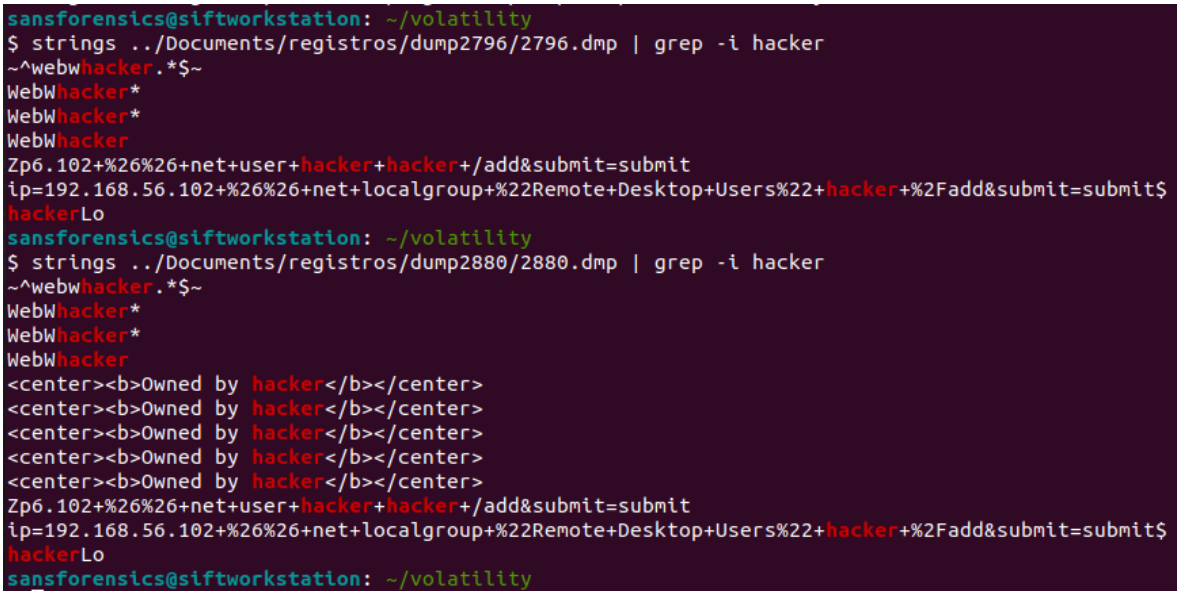

Figura 24. Análisis Memoria, procesos atacante

Fuente: Elaboración propia.

Como se puede observar encontramos la IP que vimos al realizar el análisis de los logs de XAMPP seguida de %26%26 que es el código para los caracteres "&&". Esta técnica es empleada para concatenar dos comandos y realizar inyecciones de los mismos.
Ahora que sabemos que el usuario está usando esta técnica buscamos que más comandos ha podido concretar:

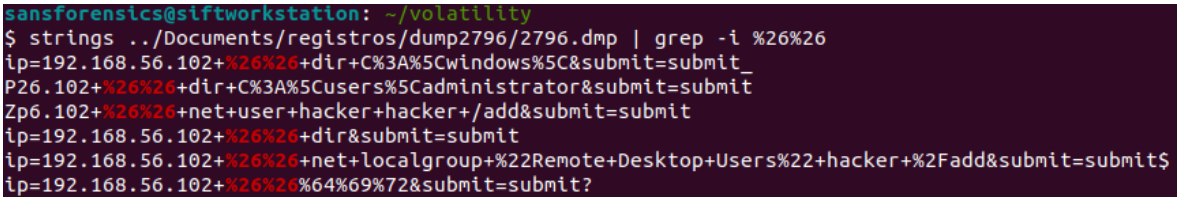

Figura 25. Análisis Memoria, procesos atacante

Fuente: Elaboración propia.

Como podemos observar son comandos para añadir al usuario "hacker" al grupo de usuarios remotos. Aun así todavía no tenemos claro al cien por cien cuál ha sido la acción que han realizado una vez han ganado acceso. Como demostramos en pasos anteriores una de las vulnerabilidades que habían estado explorando eran la de subida de archivos. Así que analizaremos las entradas de la MFT para comprobar si han podido cargar en nuestro equipo algún archivo malicioso.

4.1.3.4 NTFS Master File Table (MFT)

La MFT es un fichero en el cual están referenciados todos los archivos del NTFS (New Technology File System) que es el sistema de gestión de archivos de Windows.
Como todos los archivos están referenciados en ella, deberíamos ser capaces de encontrar los ficheros maliciosos que sospechamos que los atacantes han subido aprovechándose de las vulnerabilidades relacionadas con la subida de archivos.
Para poder analizar la MFT en primer lugar debemos extraerla de la imagen que estamos analizando. Para ello en primer lugar analizaremos la información disponible sobre la misma utilizando la herramienta "mmls". Para ello emplearemos el siguiente comando:

mmls Case1-Webserver.E01

```
root@siftworkstation:/home/sansforensics/Documents/registros# mmls Case1-Webserver.E01
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0052426751   0052424704   NTFS / exFAT (0x07)
003:  -------   0052426752   0052428799   0000002048   Unallocated
```

Figura 26. Análisis MFT, mmls

Fuente: Elaboración propia.

Con esta información ya podemos generar nuestro fichero con la MFT al cual llamaremos "MFT.raw". Para ello usaremos la herramienta "icat" mediante el siguiente comando:

icat -o 2048 Case1-Webserver.E01 0 > MFT.raw

Y por último para poder analizar e interpretar el contenido de la MFT utilizaremos la herramienta "analyzeMFT.py" con el siguiente comado:

analyzeMFT.py -f MFT.raw -o mftanalyzed.csv

Ahora que ya tenemos la MFT en formato CSV podemos analizarla en busca de los ficheros maliciosos que sospechamos que han subido los atacantes. Tras un breve análisis encontramos que en efecto los atacantes han subido al equipo distintos scripts que les permiten ejecutar comandos vía webshell.

62338	62330	Good	Active	File	1	60268	1	/xampp/htdocs/DVWA/hackable/uploads/phpshell.php
62339	62331	Good	Active	File	1	12859	4	/xampp/htdocs/DVWA/webshells.zip
62340	62332	Good	Active	Folder	1	12859	4	/xampp/htdocs/DVWA/webshells
62341	62333	Good	Active	File	1	12859	4	/xampp/htdocs/DVWA/c99.php
62342	62334	Good	Active	File	1	12859	4	/xampp/htdocs/DVWA/webshell.php
62343	62335	Good	Active	Folder	1	60268	1	/xampp/htdocs/DVWA/hackable/uploads/abc
62344	62336	Good	Active	File	1	NoParent	NoParent	NoFNRecord
62345	62337	Good	Active	File	2	60268	1	/xampp/htdocs/DVWA/hackable/uploads/phpshell2.php
62346	62338	Good	Inactive	File	2	229	2	/Users/Administrator/AppData/Local/Temp/c99 (2).php
62347	0	Zero	Inactive	File	0	Corrupt	Corrupt	Corrupt MFT Record
62348	0	Zero	Inactive	File	0	Corrupt	Corrupt	Corrupt MFT Record

Figura 27. Análisis MFT, ficheros maliciosos

Fuente: Elaboración propia.

Con estas últimas pruebas quedan ya demostrados no sólo que consiguieron acceder al servidor sino que también consiguieron aprovecharse de una serie de vulnerabilidades para introducir en él una serie de scripts maliciosos (webshells) y que usaron estos scripts para poder ejecutar comandos en el servidor.

4.1.3.5 Recopilación de sucesos descubiertos durante el análisis.

En este apartado vamos a hacer un recuento de las acciones realizadas por el atacante y a deducir como, en efecto, los ataques suelen ajustarse a un modelo. En este caso el modelo de cadena de la muerte.

- Fase de reconocimiento: Desde la IP 192.168.56.102 se realizan numerosas peticiones al servidor.
- Fase de armamento: El atacante prepara el material necesario para llevar a cabo su ataque.
- Fase de entrega:
    - Mediante un ataque de inyección se consigue crear los usuarios “user1” u “hacker” y se les incluye en el grupo de usuarios remotos.
    - Empleo de sqlmap para automatizar ataques de inyección SQL.
    - Creación de los ficheros c99.php, webshell.php, phpshell2.php, etc. Aprovechando las vulnerabilidades encontradas.
- Fase de instalación: En este caso el ataque no requería instalar ningún software, simplemente la ejecución de las webshel introducidas en el equipo.
- Fase de comando y control: Ejecución de comandos mediante la webshell.
- Fase de acciones sobre el objetivo: Una vez el atacante ya puede ejecutar comandos ya puede llevar a cabo las acciones que quiera sobre la máquina infectada.

5 Conclusiones

Este trabajo surge de mi interés en el mundo de la ciberseguridad tras mis prácticas en One eSecurity, una empresa de respuesta a incidentes. Al poco de empezar a trabajar en esta empresa empecé a interesarme en cómo se producían los ciberataques y qué formas había de detectarlos y solventarlos.

Durante el desarrollo de este trabajo y gracias a todo el proceso de documentación llevado a cabo para el mismo he podido conocer más a fondo los distintos métodos que emplean los hackers para llevar a cabo sus ataques. Esto es muy importante ya que para saber cómo defenderse y prevenir un posible ataque conviene saber primero de qué forma se llevan a cabo la mayoría de los mismos.

Con este trabajo no solo se presentan los principales tipos de ataques y malware con los que un atacante podría infectar o dañar un equipo o una red sino que también se concluye que el uso de modelos (modelo de diamante, modelo de cadena de muerte y modelo de gráfico de ataque), para analizar ciberataques puede ser de gran utilidad. Cada uno de los métodos es único y presenta los mismos ataques de formas muy diferentes. Aunque, las técnicas de modelado de ataques son diferentes entre sí, comparten algunos atributos comunes como: adversario, víctima, red, plan de ataque, crear carga útil, entrega de la instalación y ejecución de la carga útil o malware. Su uso es especialmente importante a la hora de entender los ataques, tanto el proceso que siguen como su motivación. Entender esto nos permite adelantarnos a futuros ataques y prevenirlos.

Una de las partes importantes de cualquier ciberataque es que el atacante no ataca a ninguna red de alto nivel sin llevar a cabo una investigación adecuada. El atacante planea antes de atacar. El plan se realiza utilizando datos recopilados en la infraestructura de la víctima y sus capacidades. Para realizar un ataque, el atacante debe enviar o introducir una carga útil o malware, que es una forma de ataque muy común en los equipos de trabajo, generalmente con sistema operativo Windows. Además, en ocasiones distintos métodos de ingeniería social son utilizados para entregar la carga útil a la víctima. Entonces, la probabilidad de éxito en muchas ocasiones depende en un cincuenta por ciento de la víctima, porque una vez que el archivo está entregado, depende de la víctima si ejecuta el archivo o no.

Además de todo lo anteriormente citado se realizan una serie de tres casos de prueba de detección de ataques que demuestran la importancia de los anteriores puntos a la hora de detectar, frenar y analizar un ataque una vez este ya se ha producido.

Por último concluir que aunque nunca se está completamente seguro frente a un ataque, y cuando esto sucede es de vital importancia contar con los conocimientos y las herramientas que nos permitan detectar, y solventar el ataque. Es por esto que el análisis forense digital se ha convertido en una práctica tan necesaria en la actualidad y sus profesionales son tan demandados.

Bibliografía y referencias

C. Guzmán. (2017). Protocolos para la mitigación de ciberataques en el hogar. Caso de estudio: estratos 3 y 4 de la ciudad de Bogotá. Recuperado de <https://repository.ucatolica.edu.co/bitstream/10983/15321/1/Cibersecurity%20Home.pdf.> [Accedido 12 de Enero 2021].

Centro Criptológico Nacional. (2012). Árboles de Ataque, una herramienta para la Protección de Infraestructuras Críticas. Recuperado de <https://www.ccn-cert.cni.es/seguridad-al-dia/noticias-seguridad/1238-arboles-de-ataque-una-herramienta-para-la-proteccion-de-infraestructuras-criticas.html> [Accedido 12 de Marzo 2021].

Cnsd.gob.pe. (2021) Análisis Forense Digital. Recuperado de <https://cnsd.gob.pe/index.php/gestion-de-incidentes/2-uncategorised/56-analisis-forense-digital-2> [Accedido 15 de Junio 2021].

E. Jaramillo. (2011). Técnicas de análisis forense digital aplicadas a dispositivos y sistemas móviles. Recuperado de <https://www.researchgate.net/publication/290710042_Tecnicas_de_analisis_forense_digital_aplicadas_a_dispositivos_y_sistemas_moviles>.[Accedido 20 de Enero 2021].

E. Jiménez. (2017). Los ciberataques en el marco de la responsabilidad internacional de los estados en tiempos de paz. Recuperado de <https://www.corteidh.or.cr/tablas/33456.pdf>[Accedido 14 de Febrero 2021].

European Union Agency For CyberSecurity. (2021). Man-in-the-Middle. Recuperado de <https://www.enisa.europa.eu/topics/csirts-in-europe/glossary/man-in-the-middle> [Accedido 7 de Marzo 2021].

F.J. Urueña. (2015). Ciberataques, la mayor amenaza actual. Recuperado de <http://www.ieee.es/Galerias/fichero/docs_opinion/2015/DIEEEO09-2015_AmenazaCiberataques_Fco.Uruena.pdf> [Accedido 8 de Marzo 2021].

F. Tchakounté, V.S.Nyassi, K.P. Udagepola. (2019). True Request–Fake Response: A New Trend of Spear Phishing Attack. Recuperado de <https://www.researchgate.net/profile/Kalum-Udagepola/publication/338634278_True_Request-Fake_Response_A_New_Trend_of_Spear_Phishing_Attack/links/5e4503bd92851c7f7f341eae/True-Request-Fake-Response-A-New-Trend-of-Spear-Phishing-Attack.pdf> [Accedido 24 de Marzo 2021].

J. Clarke-Salt. (2012). SQL Injection Attacks and Defense. Recuperado de <https://books.google.com/books?hl=en&lr=&id=Spm7UgBwzjIC&oi=fnd&pg=PR3&dq=sql+injection&ots=k10AZLgilD&sig=5wsm_Atpee4RQEARfbVvViCjThA#v=onepage&q=sql%20injection&f=false>[Accedido 2 de Abril 2021].

J. Olmedo. (2018). Análisis de los ciberataques realizados en América Latina. Recuperado de <https://dialnet.unirioja.es/descarga/articulo/6778118.pdf> [Accedido 5 de Marzo 2021].

K. Ryder, SANS Institute. (2011). Computer Forensics - We've Had an Incident, Who Do We Get to Investigate?. Recuperado de <https://www.sans.org/reading-room/whitepapers/incident/computer-forensics-weve-incident-investigate-652>[Accedido 20 de Enero 2021].

M. Delgado. (2007). Análisis forense digital. Recuperado de <https://www.oas.org/juridico/spanish/cyb_analisis_foren.pdf>[Accedido 28 de Diciembre 2020].

Ministerio de asuntos económicos y transformación digital, oficina de seguridad internauta. (2020). Guía de ciberataques. Recuperado de <https://www.osi.es/sites/default/files/docs/guia-ciberataques/osi-guia-ciberataques.pdf> [Accedido 5 de Marzo 2021].

N. Baliyan. (2017). Towards Improved Malware Detection using MultilevelEnsemble Supervised Learning. Recuperado de: <http://117.203.246.91:8080/jspui/handle/10266/4906>[Accedido 26 de Abril 2021]

N. Linares. (2019). Los ciberataques en el derecho internacional público. Recuperado de <https://repositori.upf.edu/bitstream/handle/10230/42136/TFGdret1819NataliaLinares.pdf?sequence=1&isAllowed=y> [Accedido 26 de Febrero 2021].

Ntfs.com. (2021). NTFS Master File Table (MFT) - NTFS.com. Recuperado de <http://ntfs.com/ntfs-mft.htm> [Accedido 10 de Junio 2021].

OpenLearn. 2021. Digital forensics. Recuperado de <https://www.open.edu/openlearn/science-maths-technology/digital-forensics/content-section-4.3> [Accedido 5 de Enero 2021].

P. Diego. (2011). Metodología de análisis forense orientada a incidentes en dispositivos móviles. Recuperado de <https://dspace.ucuenca.edu.ec/bitstream/123456789/21381/1/TIC.EC_04_Pinto.pdf> [Accedido 3 de Febrero 2021].

S. Ansen. (2017) Windows Memory Analysis with Volatility. Recuperado de <https://www.forwarddefense.com/pdfs/Memory-Analysis-with-Volatility.pdf>. [Accedido 12 de Junio 2021].

S. Ansen. (2018) Windows Event Log Analysis. Recuperado de <https://www.forwarddefense.com/pdfs/Event_Log_Analyst_Reference.pdf>. [Accedido 13 de Junio 2021].

S. Caltagirone, A. Pendergast, C. Betz. (2013). The Diamond Model of Intrusion Analysis. Recuperado de <https://apps.dtic.mil/sti/pdfs/ADA586960.pdf> [Accedido 3 de Marzo 2021].

S. Kumar. (2007). Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet. Recuperado de <https://ieeexplore.ieee.org/abstract/document/4271771/authors#authors> [Accedido 20 de Marzo 2021].

S.M. Poremba. (2017). Types of DDoS Attacks. Recuperado de <https://www.esecurityplanet.com/networks/types-of-ddos-attacks/> [Accedido 3 de Marzo 2021].

S. Raghavan. (2012). Digital forensic research: Current state of the art. Recuperado de <https://www.researchgate.net/publication/257808917_Digital_forensic_research_Current_state_of_the_art> [Accedido 10 de Diciembre 2021].

T. Yadav. (2015). Technical Aspects of Cyber Kill Chain. Recuperado de https://www.researchgate.net/publication/281148852_Technical_Aspects_of_Cyber_Kill_Chain> [Accedido 10 de Marzo 2021].

The Windows Club. (2021).What is NTUSER.DAT file in Windows 10? Recuperado de <https://www.thewindowsclub.com/ntuser-dat-file-in-windows-10>. [Accedido 13 de Junio 2021].

Threat Connect. (2014). The diamond model for intrusion analysis: A primer. Recuperado de <https://digital-forensics.sans.org/summit-archives/cti_summit2014/The_Diamond_Model_for_Intrusion_Analysis_A_Primer_Andy_Pendergast.pdf> [Accedido 10 de Marzo 2021].

V. Lo. (2014). Windows ShellBag Forensics in Depth. Recuperado de <https://www.sans.org/reading-room/whitepapers/forensics/windows-shellbag-forensics-in-depth-34545.> [Accedido 20 de Mayo 2021].

Volatility 2.6. (2016). Volatility 2.6 (Windows 10 / Server 2016). Recuperado de <https://www.volatilityfoundation.org/26> [Accedido 10 de Junio 2021].

W. Eddy. (2007). TCP SYN Flooding Attacks and Common Mitigations. Recuperado de <https://www.hjp.at/doc/rfc/rfc4987.html> [Accedido 5 de Marzo 2021].

Z. Cao. (2008). How to Launch A Birthday Attack Against DES. Recuperado de <https://eprint.iacr.org/2008/288.pdf> [Accedido 27 de Marzo 2021].

Glosario

ASP	Active Server Pages (páginas de servidores activos). Este es el primer lenguaje de programación desarrollado por la empresa estadounidense Microsoft, dicho lenguaje está orientado principalmente al lado del servidor (back end).
ASP.NET	Es un marco de trabajo el cual ofrece una serie de funcionalidades y herramientas para poder desarrollador algoritmos enfocados al back end (lado del servidor) en el cual se aplica principalmente ASP.
BGP	Border Gateway Protocol (Protocolo de puerta de enlace fronteriza).
Bot	Abreviatura de "robot" y también llamado bot de Internet: es un programa informático que funciona como agente para un usuario u otro programa, o para simular una actividad humana.
Bogon	Es un nombre informal para un paquete IP en la Internet pública que dice ser de un área del espacio de direcciones IP reservadas, pero aún no asignadas o delegadas por la Internet Assigned Numbers Authority (IANA) o por un Registro Regional de Internet delegado (RIR).
Botnet	Colección de dispositivos conectados a Internet, cada uno de los cuales ejecuta uno o más bots. Las botnets se pueden utilizar para realizar ataques de denegación de servicio distribuido (DDoS), robar datos, enviar spam y permitir que el atacante acceda al dispositivo y su conexión.
DoS	Denial-of-Service (Denegación de servicio).
DDoS	Distributed Denial-of-Service (Denegación de servicio distribuidos).
Firewall	Sistema diseñado para evitar el acceso no autorizado a una red privada o desde ella. Puede implementar un firewall en forma de hardware o software, o una combinación de ambos. Los cortafuegos evitan que usuarios de Internet no autorizados accedan a redes privadas conectadas a Internet, especialmente intranets.
Hacker	Experto en informática que utiliza sus conocimientos técnicos para lograr un objetivo o superar un obstáculo, dentro de un sistema informático por medios no estándares.
HTML	Hypertext Markup Language (Lenguaje de marco de hipertexto), se utiliza para crear documentos electrónicos (llamados páginas) que se muestran en el Internet. Cada página contiene una serie de conexiones a otras páginas llamadas

	hipervínculos. Cada página web que ve en Internet está escrita usando alguna versión de código HTML u otra.
HTTP/HTTPS	Hypertext Transfer Protocol/Secure (Protocolo de transferencia de hipertexto).
IANA	Internet Assigned Numbers Authority (Autoridad de asignación de números de Internet).
IP	Internet Protocol (Protocolo de Internet).
ICMP	Internet Control Message Protocol (Protocolo de mensajes de control de Internet).
ISP	Internet Service Provider (Proveedor de servicio de Internet).
JavaScript	Es un lenguaje de programación o secuencias de comandos que le permite implementar funciones complejas en páginas web.
J2EE	Java 2 Platform Enterprise Edition (Java 2 edición de plataforma empresarial). Es un lenguaje de programación web, orientado principalmente para el desarrollo de algoritmos de la parte back end de una plataforma web, el cual es independiente de la plataforma.
Malware	Cualquier tipo de software malicioso diseñado para dañar o explotar cualquier dispositivo, servicio o red programable.
MD	Message Digest (Resumen de mensaje).
MMS	Multimedia Messaging Service (Servicio de mensajes multimedia).
PE	Portable Executable file (archivo ejecutable portátil).
PHP	Hypertext Preprocessor (Preprocesador de hipertexto) Es un lenguaje de programación web, orientado principalmente para el desarrollo de algoritmos de la parte back end de una plataforma web.
TCP	Transmission Control Protocol (Protocolo de control de transmisión).
RAM	Random Access Memory (Memoria de acceso aleatorio).

RIR	Regional Internet Registry (Registro Regional de Internet).
SIM	Subscriber Identity Module (Módulo de identidad del suscriptor).
SMBv2	Server Message Block version 2 (Bloque de mensajes de servidor versión 2).
SMS	Short Message Service (Servicio de mensajes cortos).
SQL	Structured Query Language (Lenguaje de búsqueda estructurado).
URL	Uniform Resource Locator (localizador de recursos uniforme).